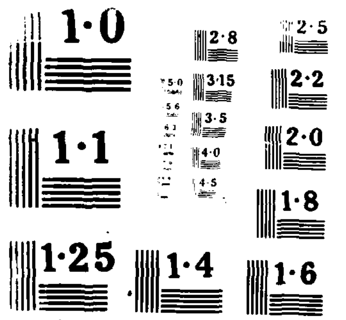
AD-A169 006 A WKB PROGRAM FOR ELF/VLF EARTH-IONOSPHERE EXCITATION BY SOURCES AT SATELLITE HEIGHTS(U) NAVAL OCEAN SYSTEMS CENTER SAN DIEGO CA R A PAPPERT ET AL. SEP 82 1/2

UNCLASSIFIED NOSC/TR-816 F/G 20/4 NL


1·0
2·8
2·5
5·0
3·15
2·2
5·6
3·5
1·1
2·0
4·0
4·5
1·8
1·25
1·4
1·6

NOSC TR 816

NOSC TR 816

AD-A169 006

Technical Report 816

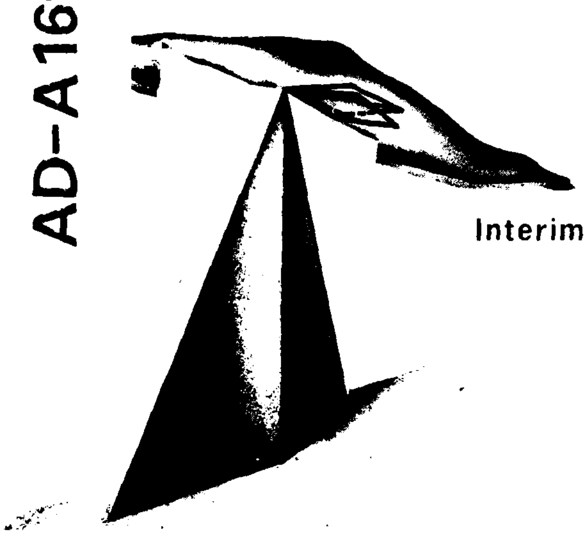
A WKB PROGRAM FOR ELF/VLF EARTH-IONOSPHERE EXCITATION BY SOURCES AT SATELLITE HEIGHTS

R. A. Pappert
L. R. Hitney

September 1982

Interim Report for Period May 1981—September 1982

Prepared for
Defense Nuclear Agency



Approved for public release; distribution unlimited

NOSC

NAVAL OCEAN SYSTEMS CENTER
San Diego, California 92152

NAVAL OCEAN SYSTEMS CENTER, SAN DIEGO, CA 92152

AN ACTIVITY OF THE NAVAL MATERIAL COMMAND

JM PATTON, CAPT, USN
Commander

HL BLOOD
Technical Director

ADMINISTRATIVE INFORMATION

This work, sponsored by the Defense Nuclear Agency under Program Element 62715H, Subtask S99QAXHB, and NOSC work unit 532-MP20A/VLF, was performed during the period May 1981 through September 1982.

Released by
JH Richter, Head
E M Propagation Division

Under authority of
JD Hightower, Head
Environmental Sciences Department

ACKNOWLEDGEMENTS

The authors are grateful to CH Shellman who developed the basic Runge-Kutta field integration program and to W DeHope for programming assistance.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: NOSC Technical Report 816 (TR 816)

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle): A WKB PROGRAM FOR ELF/VLF EARTH-IONOSPHERE EXCITATION BY SOURCES AT SATELLITE HEIGHTS

5. TYPE OF REPORT & PERIOD COVERED: Interim: May 1981-September 1982

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s):
R.A. Pappert
L.R. Hitney

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS:
Naval Ocean Systems Center
San Diego, CA 92152

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS:
62715H; S99QAXHB;
532 - MP20A/VLF

11. CONTROLLING OFFICE NAME AND ADDRESS:
Defense Nuclear Agency
Washington, D.C. 20350

12. REPORT DATE: September 1982

13. NUMBER OF PAGES: 118

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)

15. SECURITY CLASS. (of this report): Unclassified

15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

ELF (extremely low frequency)
VLF (very low frequency)
earth ionosphere wave guide
ionospheric propagation
computer programs

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report presents a variation of an earlier program for calculating ELF/VLF excitation of the earth-ionosphere waveguide by point dipoles at satellite heights. The variation consists of implementing WKB formalism developed for the purpose of speeding up the requisite field integrations. Allowance is made for the following: (1) calculating excitation factors for all electric field components E_x, E_y, and E_z; (2) calculating excitation by both electric and magnetic point dipoles; (3) calculating excitation by vertical, horizontal end-on and horizontal broadside configurations. A mode sum program is also included as an illustration of how the excitation factors are used in field calculations.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

ABSTRACT

This report presents a variation of an earlier program [3] for calculating ELF/VLF excitation of the earth-ionosphere waveguide by point dipoles at satellite heights. The variation consists of implementing WKB formalism developed by Budden [2] for the purpose of speeding up the requisite field integrations. Allowance is made for the following: (1) calculating excitation factors for all electric field components E_x, E_y and E_z; (2) calculating excitation by both electric and magnetic point dipoles; (3) calculating excitation by vertical, horizontal end-on and horizontal broadside configurations. A mode sum program is also included as an illustration of how the excitation factors are used in field calculations.

CONTENTS

I. INTRODUCTION

In a recent report [1] WKB formalism developed by Budden [2] for an anisotropic medium was used to speed up calculation of ELF/VLF modal height gains in the ionosphere to heights well in excess of the peak of the F layer. Justification for use of the WKB formalism lies in the fact that for altitudes sufficiently high in the ionosphere (i.e., $\gtrsim$ 110 km for VLF, $\gtrsim$ 150 km for ELF under daytime ionospheres and $\gtrsim$ 250 km for ELF under nighttime ionospheres) the electromagnetic field is in the outgoing whistler mode and the ionospheric gradients are sufficiently weak. In this report the WKB method is appended to an earlier program [3] for calculating excitation of the earth-ionosphere waveguide by point dipoles at satellite altitudes. As with the latter program, provision is made for calculating excitation factors for all electric field components generated by both electric and magnetic point dipoles of arbitrary orientation. It can be expected that the present program will reduce the computer cost (Univac 1100) by about a factor of 1/3 in the lower ELF band ($\simeq$75 Hz) and by more than an order of magnitude in the VLF band when the source altitudes are $\gtrsim$ 500 km. We summarize in the following paragraphs the features necessary for an understanding of the present work.

All height dependent input quantities are referenced to a Cartesian coordinate system (x, y, z) such that the z axis is taken along the vertical and positive into the ionosphere. The (x, y, z) coordinate system is referred to in this report as the input coordinate system. The transmitter and receiver are located in the x-z plane with z = 0 being the ground. The source is taken to be imbedded in a semi-infinite, homogeneous, but anisotropic slab with lower boundary termed WKBHT in the input coordinate system. Thus, reflections from above the source are ignored. The primary plane wave field radiated by the elevated antenna at an earth ionosphere eigenangle, θ_n, is decomposed into downcoming magneto-ionic components and evaluated at z = WKBHT in the input coordinate system. Next, two independent full wave starting solutions consistent with the condition of downcoming waves are assumed at an altitude termed LWSTHT in the input coordinate system. LWSTHT must be in the isotropic space between the ground and the lowest level of the ionosphere. The two independent solutions are numerically integrated upward to a level

termed TOPHT in the input coordinate system. At TOPHT the two solutions are combined to yield the proper modal polarization and decomposed into the down-coming whistler component. The full wave whistler component is then matched to the corresponding WKB component and the solution propagated to WKBHT via the WKB formulas. Finally, the strength of the full wave solution is determined by matching to the source generated whistler component previously determined at WKBHT.

The present program is a variant of a fields program [4] which necessitated starting the full wave integrations with outgoing wave and subsequent downward integration. That is in contrast with the present requirement of starting the full wave integrations with down-going wave and subsequent upward integration. To achieve this modification the specie profile and collision frequency read in (as was done in reference [3]) is left unaltered and coordinate transformations effected internally in the program. That is, read in of the latter quantities is referenced to what has been referred to previously as the input coordinate system (z axis positive into ionosphere, z = 0 at the ground). The profiles are read in from WKBHT down to level LWSTHT (or equivalently D) which is often taken to be ground (LWSTHT = 0). The coordinate system and profiles are then transformed by rotating 180° about the x-axis and translating in the z direction by the amount WKBHT. This is referred to as the prime coordinate system and represented by (x', y', z'). Mathematically the transformation is z' = -z + WKBHT. This has the effect that TOPHT' = -LWSTHT + WKBHT, LWSTHT' = -TOPHT + WKBHT and WKBHT' = 0, where the left hand side of the equations are (apart from the ') the program names after the coordinate transformation. The transformation from the (x, y, z) system to the (x', y', z') system is effected internally in the program and achieves the goal that the starting full-wave solutions in the transformed system correspond to upgoing wave and that the full wave integrations proceed as in the original full-wave program [4]. Transformation to the primed system is achieved in the subroutine XINPUT.

Basic inputs to the present program must be supplied from an ELF/VLF earth-ionosphere waveguide program such as that of reference [5]. The inputs include mode eigenangles, derivatives of the modal function evaluated at the eigenangles (or related quantities) and the elements of the plane wave ionospheric reflection matrix as defined in reference [2].

For completeness, Budden's WKB formulas [2] are summarized in section II. Formulas used in matching the full wave and WKB field are given in section III. Principal outputs of the program are the excitation factors defined in section IV. Flow of the program and subroutine descriptions are given in section V. Section VI contains a discussion of the card deck arrangement for input along with discussion of a sample output. Application of the excitation factors for calculating mode sums is discussed in section VII where sample mode sum plots are included. Appendix A contains a listing of the excitation factor program. Appendix B contains a listing of the mode sum program used to generate the results shown in section VII.

II. SUMMARY OF WKB FORMULAS

For plane wave incidence of an rf wave on the ionosphere, Maxwell's equations can be written as [6]

$$e' = -iTe \tag{1}$$

where the prime denotes $(k^{-1}\partial/\partial z)$, with k the free space wave number, T a 4 x 4 matrix given by Budden [2], and e a column matrix composed of components of the electric ($\vec{E}$) and magnetic ($\vec{H}$) fields of the rf wave. The transpose of e is

$$e^T = (E_x, -E_y, Z_o\mathcal{H}_x, Z_o\mathcal{H}_y) \tag{2}$$

where Z_o is the free space impedance. Henceforth the notation $H_x = Z_o\mathcal{H}_x$, $H_y = Z_o\mathcal{H}y$ and $H_z = Z_o\mathcal{H}_z$ will be used.

The matrix T has four characteristic roots or eigenvalues, q_i (i = 1,2,3,4), which satisfy the characteristic equation

$$\det (T - qI) = 0 \tag{3}$$

where I is the unit 4 x 4 matrix. This equation is the Booker quartic. Corresponding to any root q_i there is an eigencolumn $P = s^{(i)}$ which satisfies

$$T\, S^{(i)} = q_i s^{(i)} \tag{4}$$

Let S be the 4 x 4 matrix whose columns are $s^{(i)}$. For points where S is nonsingular, the column matrix f can be defined as

$$e = Sf \text{ or } f = S^{-1}e \tag{5}$$

and it can be shown [2] that the elements of f satisfy

$$f_k' = -iq_k f_k + \sum_j \Gamma_{kj} f_j \qquad ;j,k = 1,4 \tag{6}$$

where

$$\Gamma = -S^{-1}S' \tag{7}$$

The preceding transformation can be carried out only when S is nonsingular. When two roots of the Booker quartic are equal, two of the columns $S^{(i)}$ are usually multiples of each other, and then S is singular. Near such points some of the coupling coefficients Γ_{kj} are very large and the points may be points of reflection or points where coupling between two up-going (or downgoing) waves is very strong. The present program cannot be used in such circumstances.

When the species densities and collision frequencies vary slowly enough with height and where no two of the q_i become nearly equal, the terms of Γ are small quantities and there is an approximate solution for which the non-diagonal elements of Γ are ignored. This solution is associated with one particular root q_j of the Booker quartic. It is [2]

$$f_j = \exp\left(-ik\int^z q_j dz + k\int^z \Gamma_{jj} dz\right) \tag{8}$$

and the corresponding field components [in Budden's notation] are

$$(E_x, E_y, H_x, H_y) = (A_j F_j)^{-1/2} (a_3 q_j + a_4, -A_j, q_j A_j, a_5 q_j + a_6)$$

$$\times \exp\left(-ik\int^z q_j dz + k\int^z \Gamma_{jj} dz\right) \tag{9}$$

where

$$a_1 = -(T_{11} + T_{44}) \qquad a_4 = T_{14}T_{42} - T_{12}T_{44}$$

$$a_2 = T_{11}T_{44} - T_{14}T_{41} \qquad a_5 = T_{42} \tag{10}$$

$$a_3 = T_{12} \qquad a_6 = T_{41}T_{12} - T_{11}T_{42} \tag{11}$$

$$A_j = q_j^2 + a_1 q_j + a_2 \tag{12}$$

$$F_j = 2q_j A_j + (q_j^2 - T_{32})(2q_j + a_1) - (a_3 b_5 + b_3 a_5) \tag{13}$$

$$2\Gamma_{jj} = (A_j F_j)^{-1}\{q_j^2(a_3 b_5' - a_3' b_5 + a_5 b_3' - a_5' b_3)$$

$$+ a_4 b_6' - a_4' b_6 + a_6 b_4' - a_6' b_4 + q_j(a_3 b_6' - a_3' b_6 + a_4 b_5' - a_4' b_5 + a_5 b_4'$$

$$- a_5' b_4 + a_6 b_3' - a_6' b_3)\} \tag{14}$$

In these equations the T_{ij}'s are the elements of the T matrix given by Budden p 389.

The essence of the program documented in this report is the implementation of equation (9) for altitudes exceeding a height termed TOPHT in the input coordinate system. The mode extracted from the magneto-ionic set is the least attenuated downcoming wave. Runge-Kutta integrations are used to calculate two independent full wave solutions at TOPHT from starting conditions at LWSTHT in the input coordinate system. The full-wave solutions at TOPHT are combined to satisfy the mode polarization condition and decomposed into magneto ionic components. The downcoming whistler component is then matched to the corresponding WKB component and the fields carried to WKBHT via equation (9).

III. FIELD MATCHING

As described in the introduction, it is necessary in the primed coordinate system to integrate two independent field solutions from the level z' = TOPHT' = -LWSTHT + WKBHT to the level z'= LWSTHT' = -TOPHT + WKBHT. The initial solution vectors correspond in the prime system (at level TOPHT') to outgoing waves in vacuum and are taken to correspond to vertical and horizontal polarizations. The initial solution vectors are calculated in the subroutine WF INIT and are denoted by the matrix P as shown below

$$
\begin{array}{ll}
P(1,1) = E_x = \cos(\theta_n) = C. & P(1,2) = E_x = 0. \\
P(2,1) = -E_y = 0. & P(2,2) = -E_y = 1. \\
P(3,1) = H_x = 0. & P(3,2) = H_x = \cos(\theta_n) = C. \\
P(4,1) = H_y = 1. & P(4,2) = H_y = 0.
\end{array}
\qquad (15)
$$

where E_j and H_j represent the j^{th} component of the electric and magnetic fields of the rf wave in the prime system and θ_n represents the eigenangle for the n^{th} earth ionosphere waveguide mode. The initial solutions are integrated to the level z' = LWSTHT' via the Runge-Kutta integration scheme implemented in the subroutine WF INTG. At z' = LWSTHT' each independent full-wave solution, P_i is decomposed into four magneto-ionic components as follows

$$P_i = \sum_{j=1}^{4} \beta(i,j)\ S(j); i = 1,2 \qquad (16)$$

where S(j) are the magneto-ionic solution vectors. Thus,

$$\beta(i,j) = S^{-1}(j)P_i \qquad (17)$$

where

$$
\begin{aligned}
S^{-1}(j)S(k) &= 0 \quad j \neq k \\
&= 1 \quad j \neq k
\end{aligned}
\qquad (18)
$$

Because of the sorting achieved in the subroutine WF SORT, the upgoing (in the prime system) whistler mode is S(2). The total whistler component is structured from a linear combination of the components associated with the two independent full wave solutions. The two independent solutions are combined in such a way that they satisfy the modal polarization condition at LWSTHT in the input coordinate system

$$f = \frac{E_y}{H_y} = \frac{1 - {}_{\parallel}R_{\parallel\parallel}\bar{R}_{\parallel}}{{}_{\perp}R_{\parallel\perp}\bar{R}_{\perp}} = \frac{{}_{\parallel}R_{\perp\parallel}\bar{R}_{\parallel}}{1 - {}_{\perp}R_{\perp\perp}\bar{R}_{\perp}} \tag{19}$$

In this equation R is the plane wave reflection coefficient associated with everything above LWSTHT and $\bar{R}$ is the plane wave reflection coefficient from everything below LWSTHT in the input coordinate system. Consistent with convention, the first subscript on R or $\bar{R}$ applies to the polarization of the incident wave whereas the second subscript applies to the polarization of the reflected wave, with $_{\parallel}$ denoting vertical polarization and $_{\perp}$ denoting horizontal polarization.

The total upgoing (in the prime system) whistler component, $S_T(2)$, obtained from decomposition of the two full wave solutions at z' = LWSTHT', is then given by

$$S_T(2) = (\beta(1,2) - \beta(2,2)f)S(2) \tag{20}$$

Note that the minus sign before $\beta(2,2)$ follows from the fact that $P(2,2) = -E_y$ (see equation (15)). Equation (9) with j = 2 is normalized such that it equals equation (20) at z' = LWSTHT' and the solution propagated to z' = WKBHT' = 0 via the WKB formula. If the solution at z' = 0 is denoted by $[(\beta(1,2) - \beta(2,2)f)S(2)]_{z'=0}$, then the strength and final unknown, A_m, of the full wave solution is determined by [7, 8]

$$A_m = \frac{BP_m(I,2)e^{ikQ(2)Z'_o}}{(Q(2) - Q(1))(Q(2) - Q(3))(Q(2) - Q(4)[(\beta(1,2) - \beta(2,2)f)S(I,2)]_{Z'=0}} \tag{21}$$

where $S(I,2)(I = 1,2,3,4)$ represents the I^{th} field component of the magnetoionic mode (i.e., $I = 1 \rightarrow E_x$, $I = 2 \rightarrow -E_y$, $I = 3 \rightarrow H_x$, $I = 4 \rightarrow H_y$). Any of

the field components can be used to determine A_m. The program uses H_y or I = 4. The free space wave number is denoted by k, the Q's are Booker quartic solutions, and z'_0 is the transmitter location in the prime coordinate system. $BP_m(I,J)$ represents an amplitude factor associated with the decomposition of the primary source into an upgoing (in the prime system) magneto-ionic component (J = 2) associated with the eigenvalue θ_n. It is determined from equations (17), (22), (24), (25), and (56) of reference [8] and will not be reproduced here. The subscript m on A and BP denotes the source type and configuration in a way which will be described in the following section.

The calculations that have been described in this section are effected in the subroutine WF BNDY (or in the entry DECOMP in WF BNDY).

IV. EXCITATION FACTORS

In this section the excitation factors, λ, which have been programmed are summarized and in section VII their usage in mode sum calculations is discussed. The formulas have been derived in reference [8] and include "height gain" effects associated with transmitter and receiver. In that sense they depart from the conventional definition. The formulas programmed differ from those of reference [8] only in choice of sign and by virtue of using here the complement of the eigenangle used in reference [8]. The formulas are:

$$
\begin{aligned}
\lambda_1^m &= \frac{2CS^{1/2}}{\partial F/\partial\theta} e^{-i\pi/4}(1+{}_{\parallel}\bar{R}_{\parallel})H_y(z_R)((1-{}_{\perp}R_{\perp}{}_{\perp}\bar{R}_{\perp})h_y^m+{}_{\perp}R_{\parallel}{}_{\perp}\bar{R}_{\perp}e_y^m) \\
\lambda_2^m &= \frac{2C}{S^{1/2}\partial F/\partial\theta} e^{i3\pi/4}(1+{}_{\perp}\bar{R}_{\perp})E_y(z_R)({}_{\parallel}R_{\perp}{}_{\parallel}\bar{R}_{\parallel}h_y^m+{}_{\parallel}R_{\parallel}{}_{\parallel}\bar{R}_{\parallel}e_y^m) \\
\lambda_3^m &= \frac{2C}{S^{1/2}\partial F/\partial\theta} e^{-i\pi/4}(1+{}_{\parallel}\bar{R}_{\parallel})E_x(z_R)((1-{}_{\perp}R_{\perp}{}_{\perp}\bar{R}_{\perp})h_y^m+{}_{\perp}R_{\parallel}{}_{\perp}\bar{R}_{\perp}e_y^m)
\end{aligned}
\tag{22}
$$

In the above, F is the modal function (i.e., the determinant of $1 - R\bar{R}$). The superscript m denotes the source while the subscripts on λ denote the component of the electric field at the receiver for which the excitation factor applies. The convention governing m and j is:

$\lambda_j^m =$

j = 1 z component	m = 1 vertical electric dipole
j = 2 y component	m = 2 horizontal electric dipole broadside
j = 3 x component	m = 3 horizontal electric dipole end fire
	m = 4 vertical magnetic dipole
	m = 5 horizontal magnetic dipole broadside
	m = 6 horizontal magnetic dipole end fire

The R and $\bar{R}$ in equation (22) have been defined in the previous section. The quantities h_y^m and e_y^m are the full wave y components of the magnetic and electric fields of the rf wave and are dependent upon the type and location of

the source as indicated by the superscript m. The dependence on altitude of the receiver, z_R, in the input coordinate system is given by the functions $H_y(z_R)$, $E_y(z_R)$ and $E_x(z_R)$. A complete description of these functions is given in reference [3]. Finally, S and C are the sine and cosine, respectively, of an eigenangle, θ_n, and $\partial F/\partial\theta$ is the derivative of the mode function evaluated at θ_n.

V. PROGRAM DESCRIPTION

This section describes the subroutines in the SATELLITE program listed in appendix A. Many of the subroutines are only slight modifications of those given in reference [3]. The subroutines WKB, WKBVAR, QGAMMA and DDKXMT have been added for the purpose of implementing the WKB formalism. Principal output of the program is the excitation factors defined in section IV. A chart showing the essential structure of the SATELLITE program follows on pages 22 through 25.

SUBROUTINE MAIN

MAIN first calls for input of ionic species data in XINPUT and for computation of height gain components and excitation factors via WAVFLD. After executing WAVFLD, excitation factors are available in the array EFIELD (M,J) where M is the source designation and J the field component designation consistent with the convention given in section IV.

SUBROUTINE XINPUT (ISTART, ISTOP)

XINPUT controls read-in of input parameters via NAMELIST statements and ionic species densities and collision frequencies as a function of altitude. Common areas are set up as required. ISTART is set to 1 before the first call to XINPUT and to 0 upon subsequent calls. ISTART = 1 implies all necessary data are to be read in and ISTART = 0 signals that previously read data are to be updated, with all unspecified parameters remaining unchanged. If a value ISTOP = 0 is returned by XINPUT, then more input data are specified in the data deck for subsequent calls to WAVFLD, whereas if a value ISTOP = 1 is returned, the data read were the last data in the data deck, so that XINPUT should not be called again. XINPUT effects the transformation from the input coordinate system (x, y, z) to the prime coordinate system (x', y', z').

The data deck is divided into several parts, each of which is marked by an identifier card with the identifier DATUMFOL, PROFILE, COLLFREQ, QUIT or STOP. Each of these identifiers is described in the following section.

SUBROUTINE WAVFLD (EX, EY, EZ, HX, HY, HZ)

WAVFLD calls for Runge-Kutta integration of two independent (corresponding to outgoing wave in the prime system at TOPHT') field solutions from TOPHT' to LWSTHT' at DELHT increments. The two solutions are combined to satisfy the modal polarization condition and decomposed into magneto-ionic components at LWSTHT'. At LWSTHT' the full wave whistler component is matched to the corresponding WKB component and the solution carried to WKBHT' via WKB formulas. At WKBHT' the strength of the solution is determined for each source and orientation. WAVFLD then calls for the back substitution of normalizing values which have been saved in WF STOR. The excitation factors are then calculated in WAVFLD and placed in the array EFIELD(M,J). The field components placed at DELHT intervals in the arrays EX, EY, EZ, HX, HY, HZ are extraneous output in the present application.

SUBROUTINE WFINTG (TOPHT', LWSTHT', DELHT, IFLAG)

WFINTG performs Runge-Kutta integration of the two solution vectors in P from TOPHT' to LWSTHT'. Numerical solutions are saved at DELHT increments. Accuracy is maintained by continually adjusting the step size used in the numerical integration. The current step size (call it h) is used to obtain an estimate of P, and then a better estimate is obtained by using two integrations with step size h/2. If the two solutions agree within an error of PRECSN (an input parameter normally set to 3.D-5), the better estimate is accepted. The step size is automatically decreased to h/2 if the two estimates differ by more than PRECSN, and the integration is repeated. If the error is significantly greater than PRECSN, a step size h/2 is used. If the error is significantly less than PRECSN, the step size 2h is used if it also yields an error less than PRECSN. These tests thus form an automatic step size correction. In this program the call to WFINTG sets IFLAG equal to zero.

ENTRY INIT T

INIT T is an entry in subroutine TMTRX. INIT T sets up height independent values to be used in T matrix calculation. These include the

internally set flag ISO (ISO = 1 for isotropic calculation, 0 otherwise). Also set are the angular radio frequency, the wave number, direction cosines of the geomagnetic field, the complex sine and cosine of THETA.

SUBROUTINE TMTRX(HT')

TMTRX computes the value of the T matrix and a specific height HT' (as usual the prime simply implies that HT is referred to the prime coordinate system). The T matrix depends upon input ionospheric parameters (species density, collision frequency, angle of propagation, magnetic field, etc.). The susceptibility matrix, M, for each species in the ionosphere is computed, the effect of earth curvature is included and the T matrix is computed from the susceptibility matrix elements. The equations used to evaluate M and T are given in Clemmow and Heading [6].

SUBROUTINE WFDENS (HT', EN, COLL)

WFDENS computes the species density (returned in EN) and collision frequency (returned in COLL) at height HT' for up to five species in the ionosphere. EN and COLL are determined from corresponding profiles in the common field WFPROF. LHT and MHT are integer values which indicate which profile values are to be interpolated to find values at the height HT'. The EN (or COLL) values are given by logarithmic interpolation of the profile values of heights ENHT(LHT + 1) and ENHT(LHT) or (COLLHT(MHT + 1) and COLLHT(MHT)).

SUBROUTINE WF INIT(P)

WF INIT sets the two starting field vectors P, at TOPHT' subject to the condition of outgoing wave in the prime system. The two starting solutions correspond to TE and TM polarizations.

SUBROUTINE PDERIV(P,DPDH)

PDERIV computes the height derivatives of the two field vectors in P according to Clemmow and Heading [6]. The derivative is returned in DPDH.

SUBROUTINE XFER (A, B, M)

Transfers the N element array A into B.

ENTRY WF STOR

This is an entry in WFSCAL where certain values obtained during integration through the ionosphere are saved for later use. The solution matrix P, the height for which P is a solution and a height integer index are saved along with orthogonalization and normalization values OSUM, APROD and BPROD. In addition, values of the susceptibility matrix elements M31, M32 and M33, which are needed to compute the EZ component of the electric field, are saved at each height.

SUBROUTINE WF STEP (P,DPDH, HT', DELHT, IFLAG)

WF STEP numerically advances the solution matrix P, using the derivative DPDH, from HT' to HT'-DELHT by the Runge-Kutta method. IFLAG, set internally, controls the calculations at intermediate points between HT' and HT'-DELHT at which evaluations are required for comparison of the second and fourth order Runge-Kutta integrations.

SUBROUTINE WF SCAL (PP, IFLAG)

WF SCAL normalizes and orthogonalizes the solution vectors PP according to the formulas of reference [4]. This scaling must later be removed to yield correct unscaled solutions. Control for calculating the quantities needed for removal of the scaling is the internally set IFLAG. Calling WF SCAL sets IFLAG = 0 in this program.

ENTRY DECOMP(HT')

This is an entry in WF BNDY. The two full wave solution vectors, P, are decomposed at HT' = LWSTHT' into upgoing (in the prime system) magneto-ionic components and combined to satisfy the modal polarization condition given in equation (19).

SUBROUTINE EIGVAL (A, SIG)

EIGVAL computes the eigenvalues (returned in SIG) or the arbitrary 4×4 complex matrix A. The characteristic polynomial $P(\lambda)$ of A is determined by explicitly computing $\det(A-\lambda I) = P(\lambda)$. The roots of this quartic polynomial (as computed in closed form, see subroutine QUARTC) are the desired eigenvalues of A. If the ionosphere is isotropic, the eigenvalues of A are computed as the roots of two quadratics.

SUBROUTINE QUARTC (FOUR B3, SIX B2, FOUR B1, B0,Q)

QUARTC computes the four roots of the polynomial Q^4 + FOUR B3 * Q^3 + SIX B2 * Q^2 + FOUR B1 * Q + B0 in closed form [9]. Up to ten applications of Newton's iteration are then performed to improve the accuracy of the roots, if necessary.

SUBROUTINE EIGVEC (A, SIG, VEC)

EIGVEC computes the four eigenvectors of A (returned in the four columns of the matrix VEC). The eigenvalues of A are assumed given in SIG. Each eigenvector X_i is computed as the solution of $(A - \sigma_i I)X_i = 0$. Since $\det(A - \sigma_i I) = 0$, this system has a solution if one element of X_i is chosen arbitrarily. In order to reduce the special isotropic case, the first element of X_1 and X_4 are arbitrarily set to A_{23}. The other three elements of each vector are then determined as the solution of a linear system in three unknowns. If the ionosphere is isotropic, the eigenvectors (corresponding to the quadratic roots described previously) are computed in simplified form.

SUBROUTINE WF SORT (Q, A IFAIL)

WF SORT arranges the eigenvalues Q and eigenvectors A of the Booker quartic so that they occur in the order of upgoing fast (evanescent), upgoing slow (travelling or whistler component), downgoing slow, and downgoing fast in the prime coordinate system. If the eigenvalues are too large in magnitude or if they cannot be ordered IFAIL is set to 1 and an error message results.

SUBROUTINE CIMVER (A, AINV, N, NDIM, ERR)

CINVER computes the inverse (returned in AINV) of the N×N matrix A. NDIM is an integer variable which must be greater than or equal to N. ERR is the estimated relative error of the inverse matrix.

SUBROUTINE CLINEQ (A, B, X, N, NDIM, IFLAG, ERR)

CLINEQ computes the solution of simultaneous linear equations with complex coefficients. That is, it solves the matrix A * X = B for the vector X of length N, given the matrix A of size N by N and the vector B of length N. The matrix A is destroyed by CLINEQ. NDIM is an integer variable which must be greater than or equal to N. IFLAG is an integer variable normally set to 0. ERR indicates the relative error in the computed solution of vector X.

SUBROUTINE RBARS (C, S, RBAR11, RBAR22)

RBARS calculates the plane wave reflection coefficients $_{\parallel}\bar{R}_{\parallel} \doteq$RBAR11 and $_{\perp}\bar{R}_{\perp} \doteq$RBAR22 looking towards the ground from LWSTHT in the input system. Evaluations are in terms of modified Hankel functions of order 1/3. The cosine (C) and sine (S) of the eigenvalue, THETA, are used in the $\bar{R}$ determination.

SUBROUTINE MDHNKL (Z, H1, H2, H1PRME, H2PRME)

MDHNKL calculates for argument Z two independent solutions (H1 and H2) and their derivatives (H1PRME, H2PRME) of Stokes' equation by methods described in reference [10]. H1 and H2 are the modified Hankel functions of order 1/3 referred to above.

SUBROUTINE WKB

If STPFLG.NE.1, WKB extends calcuation of the solution vector from LWSTHT' to WKBHT' by means of the formulas in section II. Integration of the phase factors q_j and Γ_{jj} (j index for upgoing, in prime system, whistler mode) is performed by Simpson rule routine with fixed step size that can be either input to the program or set internally.

SUBROUTINE WKVAR

If STPFLG.EQ.1, WKBVAR extends calculation of the solution vector from LWSHT' to WKBHT' via the WKB formulas of section II. Although generally requiring more CAU time this is the preferred option since it involves checks on step size. Integration of the phase factors q_j and Γ_{jj} (j index for upgoing, in prime system, whistler mode) is performed by a Simpson rule routine which maintains precision by adjustment of the step size much like the Runge-Kutta step size adjustment in WFINTG.

ENTRY INITDT

This is an entry in DDKXMT where all height independent quantities are initialized before extending field calculations from LWSTHT' to WKBHT' via the WKB method.

SUBROUTINE Q GAMMA (HT', DELHT, LWSTHT', WKBHT', Q, GAMMA)

Q GAMMA determines the Booker quartic solutions for the least attenuated upgoing (in the prime system) magneto-ionic component at LWSTHT' and computes the a_i's, b_i's, a_i''s, b_i''s, A_i, F_i and Γ_{ii} according to formulas of section II. Full-wave solutions are matched at LWSTHT' to the WKB solutions. At other heights and up to WKBHT' coefficients of the exponentials in equation (9) of section II are calculated along with coefficients required for calculating the EZ and HZ fields.

SUBROUTINE DDKXMT

DDKXMT calculates susceptibility matrix elements, M, the T matrix elements and their derivatives with respect to height in the height range LWSTHT' $\leq$ HT' $\leq$ WKBHT'.

WF BNDY(B)

This computes coefficients $BP_m(I,2)$--see equation (21)--for each source and orientation at WKBHT'. The coefficients are used to determine the strength of the full wave solution which has been carried from within the guide to WKBHT' by the combination of full wave and WKB methods.

ENTRY WF LOAD

WF LOAD is an entry in WF SCAL which makes available for the purpose of back substitution into the full wave solutions the orthogonalizing and normalizing coefficients saved in WF STOR.

ENTRY HT GAIN

This is an entry in RBARS where the quantities required to calculate the properly normalized height gains at the receiver altitude are evaluated.

MAIN

Calls for input data via XINPUT. Calls for computation by WAVFLD of excitation factors for the ELF/VLF electric field components beneath the ionosphere produced by electric and magnetic dipoles at satellite altitudes.

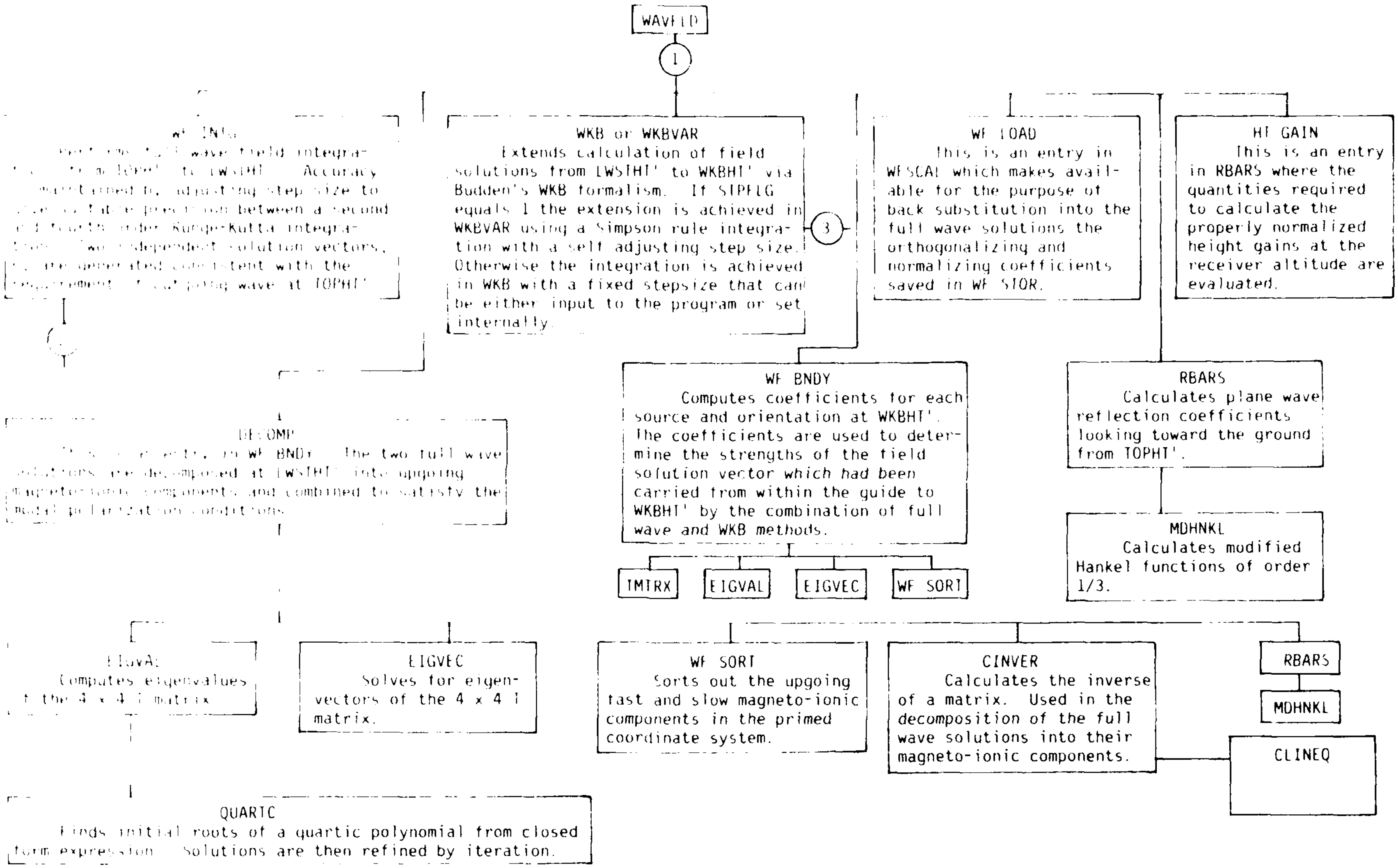
XINPUT

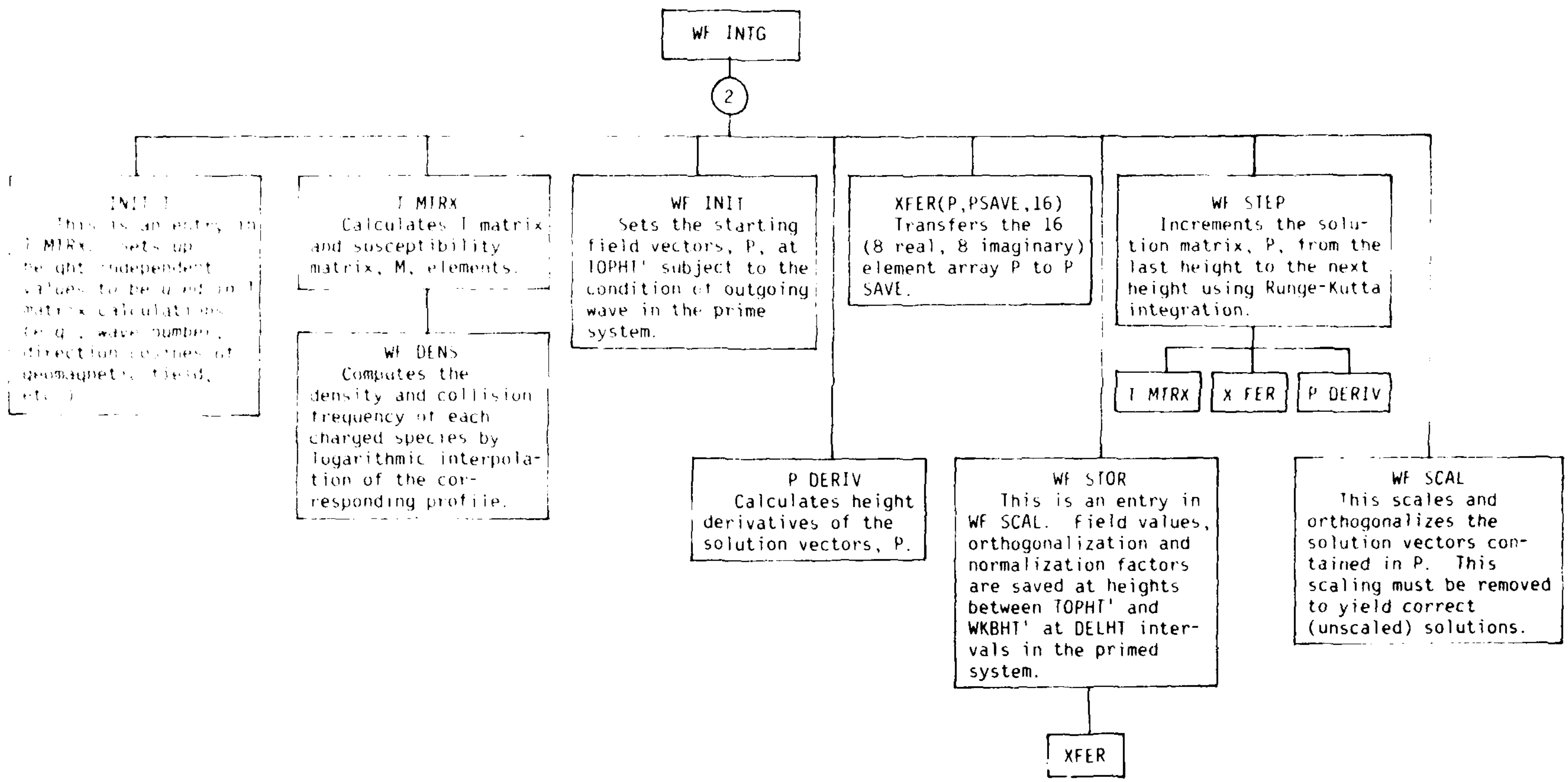
Input electron and ion species densities and collision frequencies as a function of altitude in the input coordinate system (z positive into the ionosphere). Converts input to the primed coordinate system (z' positive towards the ground).

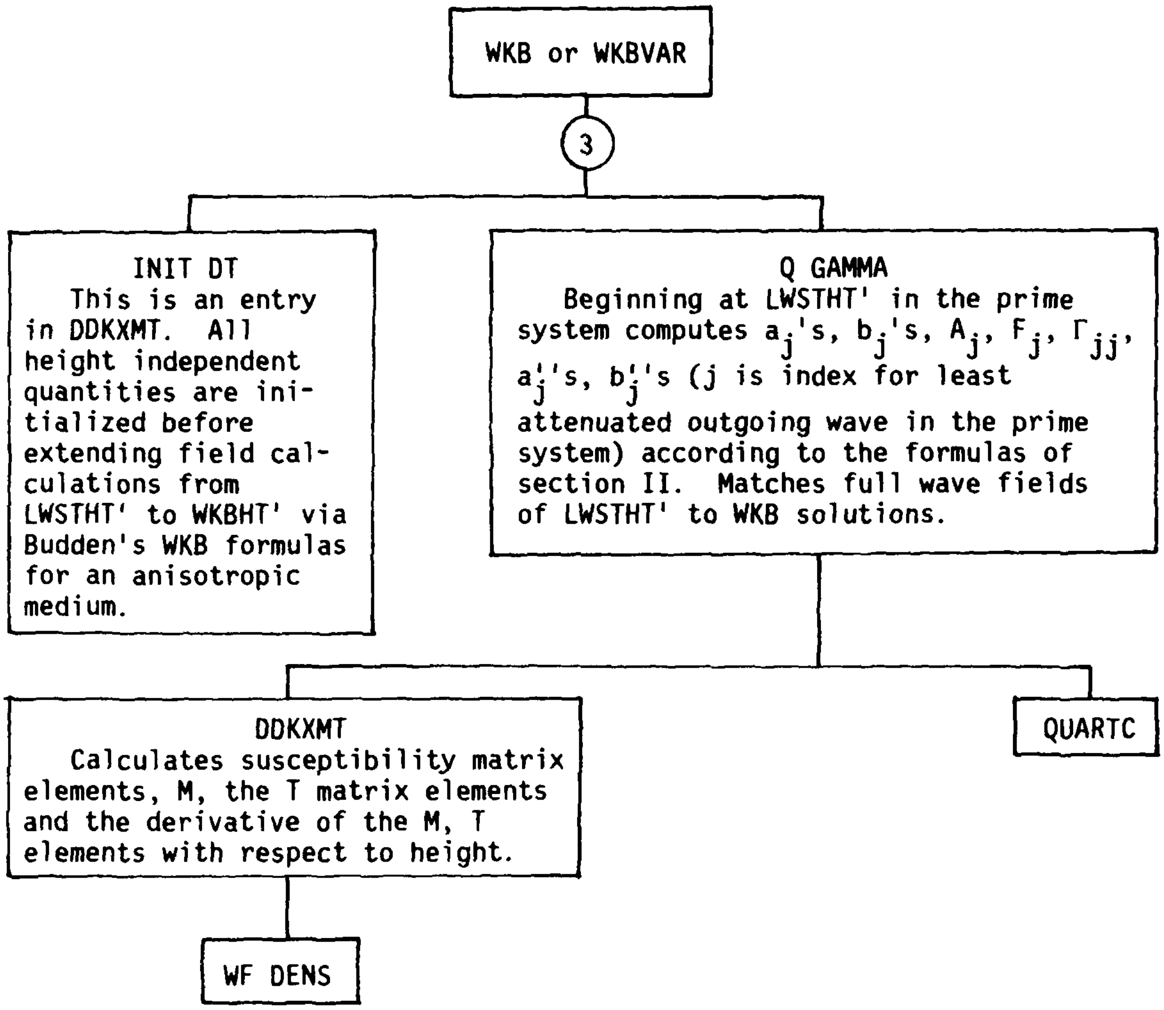
WAVFLD

Calls for Runge-Kutta integration of two independent field solutions from TOPHT' to LWSTHT' at DELHT increments. The two solutions are combined to satisfy the modal polarization condition and decomposed into magneto-ionic components at LWSTHT'. WAVFLD calls for the back substitution of orthogonalizing and normalizing values (saved as data in WFSTOR). It calls for the extension of the fields from LWSTHT' to WKBHT' via Budden's WKB formalism. At WKBHT' the field components are matched to the corresponding components generated by each source. Excitation factors for the ELF/VLF electric field components are then computed for an input receiver altitude which must be within the isotropic space between the ground and ionosphere.

1


WAVFLD
1
WKB or WKBVAR
Extends calculation of field solutions from LWSTHT' to WKBHT' via Budden's WKB formalism. If SIPFLG equals 1 the extension is achieved in WKBVAR using a Simpson rule integration with a self adjusting step size. Otherwise the integration is achieved in WKB with a fixed stepsize that can be either input to the program or set internally.
3
WF LOAD
This is an entry in WFSCAL which makes available for the purpose of back substitution into the full wave solutions the orthogonalizing and normalizing coefficients saved in WF STOR.
HT GAIN
This is an entry in RBARS where the quantities required to calculate the properly normalized height gains at the receiver altitude are evaluated.
WF BNDY
Computes coefficients for each source and orientation at WKBHT'. The coefficients are used to determine the strengths of the field solution vector which had been carried from within the guide to WKBHT' by the combination of full wave and WKB methods.
TMTRX
EIGVAL
EIGVEC
WF SORT
RBARS
Calculates plane wave reflection coefficients looking toward the ground from TOPHT'.
MDHNKL
Calculates modified Hankel functions of order 1/3.
EIGVEC
Solves for eigenvectors of the 4 x 4 T matrix.
WF SORT
Sorts out the upgoing fast and slow magneto-ionic components in the primed coordinate system.
CINVER
Calculates the inverse of a matrix. Used in the decomposition of the full wave solutions into their magneto-ionic components.
RBARS
MDHNKL
CLINEQ
QUARTIC
Finds initial roots of a quartic polynomial from closed form expression. Solutions are then refined by iteration.


WF INTG
2
I MTRX
Calculates I matrix and susceptibility matrix, M, elements.
WF DENS
Computes the density and collision frequency of each charged species by logarithmic interpolation of the corresponding profile.
WF INIT
Sets the starting field vectors, P, at TOPHT' subject to the condition of outgoing wave in the prime system.
XFER(P,PSAVE,16)
Transfers the 16 (8 real, 8 imaginary) element array P to P SAVE.
WF STEP
Increments the solution matrix, P, from the last height to the next height using Runge-Kutta integration.
I MTRX
X FER
P DERIV
P DERIV
Calculates height derivatives of the solution vectors, P.
WF STOR
This is an entry in WF SCAL. Field values, orthogonalization and normalization factors are saved at heights between TOPHT' and WKBHT' at DELHT intervals in the primed system.
XFER
WF SCAL
This scales and orthogonalizes the solution vectors contained in P. This scaling must be removed to yield correct (unscaled) solutions.


WKB or WKBVAR
3
INIT DT
This is an entry in DDKXMT. All height independent quantities are initialized before extending field calculations from LWSTHT' to WKBHT' via Budden's WKB formulas for an anisotropic medium.
Q GAMMA
Beginning at LWSTHT' in the prime system computes a_j's, b_j's, A_j, F_j, Γ_{jj}, a_j''s, b_j''s (j is index for least attenuated outgoing wave in the prime system) according to the formulas of section II. Matches full wave fields of LWSTHT' to WKB solutions.
DDKXMT
Calculates susceptibility matrix elements, M, the T matrix elements and the derivative of the M, T elements with respect to height.
QUARTC
WF DENS

VI. SAMPLE INPUT AND OUTPUT

A. INPUT

Altitude independent parameters, species densities and collision frequencies as a function of altitude are supplied via an input data deck on a standard input unit. Read-in occurs in the subroutine XINPUT. The data deck is divided into several parts, each of which is marked by control cards DATUMFOL, PROFILE, COLLFREQ, QUIT, or STOP. Each of these control cards is described below:

(1) DATUMFOL

DATUMFOL - is a control card signifying that input for the namelist DATUM follows.

&DATUM - initiates reading of input data which do not vary with altitude. The data is read in namelist input format. All altitude related quantities are referenced to the input coordinate system (z positive into the ionosphere, ground at z = 0). These data are:

THETA - complex angle of incidence in degrees as measured at height H. THETA must be applied from a waveguide program (e.g., ref. [5]).

FREQ - frequency in kHz.

TOPHT - height to which full wave integrations are carried (km).

LWSTHT - beginning height for full wave integrations (km). Also equivalent to level D where mode equation is evaluated.

WKBHT - lower boundary of homogeneous, anisotropic slab in which the transmitter is imbedded (km). Also height to which field solution is propagated via WKB formalism from TOPHT.

DELHT - height increment in km's at which field solutions are saved.

PRECSN - accuracy to be maintained locally in the numerical integrations. Usually taken to be the default value of 3.0E-5.

AZIM - azimuth of propagation path in degrees, measured clockwise from geomagnetic north.

CODIP - codip of geomagnetic field in degrees.

MAGFLD - geomagnetic field strength in webers/square meter.

COEFNU(5)- coefficient of exponential form of collision frequency (if not specified by profile). Up to five values may be specified. One for each species.

EXPNU(5) - exponent of exponential form of collision frequency (if not specified by profile). Up to five values may be specified. One for each species.

ALPHA - earth curvature coefficient in inverse km. Default is 3.14E-4.

SIGMA - ground conductivity in siemens/meter. Default value is seawater value of 4.64.

EPSLON - ground permittivity in farads/meter. Default value is 7.172015D-10. This corresponds to a dielectric constant of 81 for seawater.

EPSLNO - permittivity of free space (8.8544E-12) in farads/meter.

H - altitude in km at which modified refractive index is unity. Eigenangles are referenced to this altitude. H must be consistent with the H setting in the waveguide program which supplied the eigenangles.

TXHT - transmitter altitude (km).

RXHT - receiver altitude (km).

ITR - integer flag which should be set to zero in this program.

RMAG(4) - magnitude of the plane wave reflection elements of R. $R(1) \doteq {}_{\parallel}R_{\parallel}$, $R(2) \doteq {}_{\perp}R_{\perp}$, $R(3) = {}_{\perp}R_{\parallel}$, $R(4) = {}_{\parallel}R_{\perp}$. These elements are referenced to level D (or equivalently LWSTHT) in the input coordinate system and must be supplied from a waveguide program such as that of reference [5].

RANG(4) - phase of the plane wave reflection elements of R. These elements are referenced to level D (or equivalently LWSTHT) in the input coordinate system and must be supplied from a waveguide program such as that of reference [5].

XTRMAG - magnitude of the ground based vertical dipole excitation factor for the vertical electric field, Ez. This, along with XTRANG, is used for the purpose of obtaining the derivative of the modal equation, $dF/d\theta$. XTRMAG must be supplied from a waveguide program such as that of reference [5].

XTRANG - phase of the ground based vertical dipole excitation factor for the vertical electric field, Ez. This, along with XTRMAG, is used for the purpose of obtaining the derivative of the modal equation, $dF/d\theta$. XTRANG must be supplied from a waveguide program such as that of reference [5].

NUMDIV - number of divisions into which DELHT is divided when using subroutine WKB (i.e., when STPFLG $\neq$ 1) with fixed input step size. NUMDIV must then be a multiple of 2. If STPFLG $\neq$ 1 and NUMDIV = 0 then the program will determine a constant step size to be used with the Simpson rule integration. It is repeated here that the preferred option is with STPFLG = 1 because of the checks on the adequacy of the step size.

STPFLG - integer flag which must be set to 1 if the recommended WKBVAR subroutine is to be used to effect the WKB integrations. Other options are discussed above (NUMDIV).

NRSPEC - number (integer) of species in the ionosphere. Can take on values up to 5. Default value is 1.

CHARGE(5)- sign of charge of each species in proton units. For an electron, the charge is -1.0. Default values are (-1.0, 1.0, -1.0, 1.0, -1.0).

RATIOM(5)- mass of each species relative to mass of an electron. Default values are (1.0, 5.8D4, 5.8D4, 5.8D4, 5.8D4).

&END - signifies end of the DATUM namelist input.

(ii) PROFILE

PROFILE - is a control card initiating reading of the ionospheric profile cards. All altitudes and related quantities are referenced to the input coordinate system (z positive into the ionosphere, ground at z = 0). The control card PROFILE is followed by an alpha-numeric card which is used to identify the profile. The profile is input starting at the top of the ionosphere (i.e., WKBHT). The cards must be input in descending order by height. The profile cards contain the height in kilometers and the species densities in particles per cubic centimeter. A maximum of five species can be specified. In the special case of three species, only two are specified on the card. The first is assumed to be electrons and the second is positive ions. The third species, negative ions, is calculated by subtracting the electron density from the positive ion density. All three species are listed on the computer printout. If the value of any species density is less than or equal to zero, it is set in the program to 1.0E-40. The heights are punched in the form xxx.xx with the decimal point in column 5, and the densities are punched in the form x.xxD±xx with the decimal points in column 15, 25, etc. All species data must be specified except for the special case discussed above. The end of the profile is indicated by a dummy height of 999.99.

(iii) COLLFREQ

COLLFREQ - is a control card initiating reading of the collision frequency profile. All altitudes and related quantities are referenced to the input coordinate system (z positive into the ionosphere, ground at z = 0). The control card COLLFREQ allows for using nonexponential collision frequencies. A strictly exponential collision frequency may be specified in namelist input by the variables COEFNU and EXPNU. If a collision frequency profile deck is used it overrides COEFNU and EXPNU. Collision frequencies for all species must be input. The card preparation is just as above with the species density values replaced by collision frequency values (in collisions/sec) on the cards. The end of the profile is indicated by a dummy height of 999.99.

(iv) QUIT

QUIT - is a control card which indicates the end of input data for a call to WAVFLD. After calling WAVFLD, XINPUT can be read and used as data for a subsequent call to WAVFLD. This allows several runs to be made with one input deck. Note that only those data which are changed need be specified after the card QUIT.

(v) STOP

STOP - control card which indicates that there are no more input data and that the run is to be terminated after the next call to WAVFLD. Note that if QUIT is encountered ISTOP is set to zero, but if STOP is encountered ISTOP is set to one.

A schematic of an input deck is shown on page 34, and one actual sample input is shown on pages 35 through 37.

B. OUTPUT

The output shown on pages 38 through 41 corresponds to the first DATUMFOL shown on page 36. The output corresponds to the STPFLG = 1 setting. Output corresponding to the additional DATUMFOL have not been shown simply to cut down on the number of pages.

First come NAMELIST and profile printouts in the input coordinate system (z positive into the ionosphere, ground at z = 0). Natural logarithms of the profiles in the prime system are also printed. Next follows output which gives the number of Runge-Kutta integration steps used in WAVFLD during the course of integrating from TOPHT' to LWSTHT' as well as the smallest and the average step sizes used in the WKB extension from LWSTHT' to WKBHT'. This is followed by the principal output; two nine element arrays which are the excitation factors for both electric and magnetic dipole transmitters. The column headings indicate excitation factors for the z, y and x electric field components in the input coordinate system. The row labels indicate excitation by vertical (V), horizontal broadside (HB) and horizontal end-on (HE) point dipoles of either electric or magnetic type.

We list below a comparison at 20 kHz between the excitation factors obtained using full wave (FW) methods up to 500 km as opposed to using WKB formalism between 125 km and 500 km. The agreement will be seen to be excellent.

ELECTRIC DIPOLE TRANSMITTER

		Z MAG	Z ANG (RAD.)	Y MAG	Y ANG (RAD.)	X MAG	X ANG (RAD.)
V	WKB	.913-4	-1.504	.118-7	-2.234	.460-7	2.425
	FW	.913-4	-1.485	-.118-7	-2.215	.460-7	2.443
HB	WKB	.809-4	1.358	.104-7	.628	.408-7	-.996
	FW	.808-4	1.377	.104-7	.647	.407-7	-.978
HE	WKB	.816-4	-.243	.105-7	-.973	.412-7	-2.597
	FW	.816-4	- 224	.105-7	-.954	.411-7	-2.579

MAGNETIC DIPOLE TRANSMITTER

		Z MAG	Z ANG (RAD.)	Y MAG	Y ANG (RAD.)	X MAG	X ANG (RAD.)
V	WKB	.785-4	2.928	.101-7	2.198	.396-7	.573
	FW	.785-4	2.946	.101-7	2.216	.396-7	.591
HB	WKB	.279-2	1.358	.360-6	.628	.141-5	-.996
	FW	.279-2	1.377	.360-6	.647	.141-5	-.978
HE	WKB	.279-2	-.212	.360-6	-.942	.141-5	-2.567
	FW	.279-2	-.194	.360-6	-.924	.141-5	-2.549

EXAMPLE OF INPUT DECK

```
DATUMFØL

&DATUM
(input data)
&END




PRØFILE

(profile cards for each specie)

999.99

CØLLFREQ

(coll. frequency cards for each specie)

999.99

QUIT (end of input for this run)

DATUMFØL

&DATUM
(changes in input data)
&END

PRØFILE

(specify entire new profile deck)

999.99

QUIT (end of input for this run)

DATUMFØL

&DATUM
(changes in input data)
&END

STØP (end of data deck)
```

SAMPLE INPUT

```
DATUMFOL
 &DATUM
 THETA=189.300,-6.297),
 RMAG=.45537,.48488,.75127,.71191,
 RANG=39.23390,325.36661,256.47889,265.18541,
 XTRMAG=2.2393E-3,XTRANG=1.755,
 FREQ=20.0,
 IDG=2,
 TOPHT=125.,LWSTHT=0.,WKBHT=500.,DELHT=25.,
 PRECSN=.3E-4,
 AZIM=238.5,CDIP=39.0,MAGFLD=4.31E-5,
 SIGMA=4.64,EPSLON=7.172015E-10,
 EPSINO=.885,34D-11,
 TXHT=500.,RXHT=0.,
 H=0.,
 NUMDIV=0,
 STPFLG=1,
 NRSPEC=1,
 &END
PROFILE 1
GLOBAL NIGHTTIME IONOSPHERE  (SAT. NIGHT ABOVE 99, H'=87 BELOW 99)
 500.00       2.00D+05
 400.00       4.00D+05
 370.00       4.30D+05
 350.00       4.40D+05
 330.00       4.30D+05
 250.00       1.00D+05
 225.00       5.00D+03
 220.00       3.20D+03
 210.00       1.68D+03
 200.00       1.00D+03
 190.00       6.50D+02
 180.00       4.60D+02
 170.00       3.45D+02
 160.00       2.82D+02
 155.00       2.60D+02
 150.00       2.50D+02
 145.00       2.55D+02
 140.00       2.80D+02
 130.00       3.70D+02
 120.00       5.80D+02
 112.00       1.10D+03
 110.00       1.30D+03
 106.00       1.70D+03
 104.00       1.90D+03
 102.00       1.96D+03
 100.00       2.00D+03
  99.00       1.95D+03
   0.00       1.83D-12
 999.00
COLLFREQ
 500.00       2.00D+02
 400.00       3.80D+02
 370.00       3.90D+02
```

SAMPLE INPUT

```
      390.00      3.80D+02
      330.00      3.50D+02
      300.00      2.80D+02
      250.00      1.05D+02
      225.00      3.50D+01
      220.00      3.00D+01
      210.00      3.30D+01
      200.00      4.50D+01
      150.00      1.60D+03
      120.00      1.00D+04
      104.00      3.00D+04
        0.00      1.82D+11
      999.99
    QUIT
    DATUMFOL
     &DATUM
     THETA=(89.734,-4.965),
     RMAG=.37698,.50137,.70502,.66571,
     RANG=340.64034,267.69431,207.14367,215.38363,
     XTRMAG=1.3353E-2,XTRANG=1.832,
     PRECSN=.3E-4,
     &END
    QUIT
    DATUMFOL
     &DATUM
     THETA=(86.562,-0.602),
     RMAG=.31415,.51674,.70714,.66251,
     RANG=255.51615,179.85049,128.19374,135.23562,
     XTRMAG=4.7435E-2,XTRANG=1.553,
     PRECSN=.3E-4,
     &END
    QUIT
    DATUMFOL
     &DATUM
     THETA=(83.728,-0.339),
     RMAG=.25640,.48657,.70436,.65576,
     RANG=211.90089,126.88680,81.77360,87.87696,
     XTRMAG=2.3667E-2,XTRANG=1.762,
     PRECSN=.3E-4,
     &END
    QUIT
    DATUMFOL
     &DATUM
     THETA=(79.891,-0.477),
     RMAG=.15081,.49026,.78613,.72446,
     RANG=145.66026,22.54322,355.24173,359.06332,
     XTRMAG=3.5551E-2,XTRANG=1.414,
     PRECSN=.3E-4,
     &END
    QUIT
    DATUMFOL
     &DATUM
     THETA=(77.714,-0.346),
     RMAG=.10293,.45046,.74700,.68147,
     RANG=131.65558,312.34340,298.43223,300.44511,
     XTRMAG=2.7565E-2,XTRANG=1.854,
     PRECSN=.3E-4,
```

SAMPLE INPUT

```
  &END
 QUIT
 DATUMFOL
  &DATUM
  THETA=(74.296,-0.500),
  RMAG=.16695,.52330,.77911,.70326,
  RANG=98.50531,198.50546,203.54697,201.65665,
  XTRMAG=2.4133E-2,XTRANG=1.212,
  PRECSN=.3E-4,
  &END
 QUIT
 DATUMFOL
  &DATUM
  THETA=(72.241,-0.442),
  RMAG=.22314,.53869,.73045,.65456,
  RANG=52.33535,130.93816,144.63344,139.79135,
  XTRMAG=3.3945E-2,XTRANG=1.887,
  PRECSN=.3E-4,
  &END
 STOP
```

SAMPLE OUTPUT

```
DATUMFOL
 $DATUM
THETA =( .8979[illegible]+002,-.62[illegible]+001),FREQ = .2[illegible]+002,IDRS =            2,
TCPHT = .125[illegible]+003,[illegible] = .0000000000000000    ,[illegible] = .5[illegible]+003,[illegible] = .2500000000000000+002,
PRECSN = .30[illegible]-004,AZIM = .2[illegible]+003,[illegible] = .3[illegible]+002,[illegible] = .4309999[illegible]999-004,
COEFNU = .18[illegible]+012, .0[illegible]     , .0[illegible]     , .0[illegible]0     , .0000000000000000
EXPNU =-.15[illegible]-003, .0[illegible]     , .0[illegible]     , .0[illegible]     , .0000000000000000     ,
ALPHA = .3[illegible]-00[illegible],SIGMA = .4[illegible]+001,EPSLON = .7172015000[illegible]-0[illegible],EPSNO = .8854339[illegible]99999997-011,
TXHT = .50[illegible]+003,RXHT = .0000000000000000    ,H = [illegible]000000000000     ,
RMAG = .45337[illegible]+000, .4[illegible]+000, .751[illegible]+000, .71[illegible]+000,
RANG = .392339[illegible]+002, .325[illegible]+003, .25[illegible]+003, .205[illegible]99+003,
XIRMAG = .22392[illegible]-002,XTRANG = .175[illegible]+001,NUMDIV =          0,STPFLG =          1,NRSPEC =          1,
CHARGE =-.10000000[illegible]+001, .1000000000000000+001,-.1000000000000000+001, .1000000000000000+001,-.1000000000000000+001
RATIOM = .1000000000000000+001, .5799999999999999+005, .5799999999999999+005, .5799999999999999+005, .5799999999999999+005
$END
PROFILE 1
GLOBAL NIGHTTIME IONOSPHERE  (SAT. NIGHT ABOVE 99, H'=87 BELOW 99)
 500.00      2.00+005
 400.00      4.00+005
 370.00      4.30+005
 350.00      4.40+005
 3[illegible]0.00      4.30+005
 250.00      1.00+005
 225.00      5.00+003
 220.00      3.20+003
 210.00      1.66+003
 200.00      1.00+003
 190.00      6.50+002
 180.00      4.60+002
 170.00      3.45+002
 160.00      2.82+002
 155.00      2.60+002
 150.00      2.50+002
 145.00      2.55+002
 140.00      2.80+002
 130.00      3.70+002
 120.00      5.80+002
 112.00      1.10+003
 110.00      1.30+003
 106.00      1.70+003
 104.00      1.90+003
 102.00      1.94+003
 100.00      2.00+003
  99.00      1.95+003
    .00      1.83-012
 999.9[illegible]
 5[illegible].00     -2.70+00[illegible]
 4[illegible]1.00      7.59+0[illegible]
 400.00      7.6[illegible]+00[illegible]
 3[illegible]8.00      7.59+0[illegible]
 3[illegible]5.00      7.55+0[illegible]
 3[illegible]4.00      7.44+000
 3[illegible]0.00      7.17+000
 398.00      7.00+000
```

SAMPLE OUTPUT

```
 380.00     6.36+000
 370.00     5.91+000
 360.00     5.65+000
 355.00     5.54+000
 350.00     5.52+000
 345.00     5.56+000
 340.00     5.64+000
 330.00     5.84+000
 320.00     6.13+000
 310.00     6.48+000
 300.00     6.91+000
 290.00     7.43+000
 280.00     8.07+000
 275.00     8.52+000
 250.00     1.15+001
 170.00     1.30+001
 150.00     1.30+001
 130.00     1.30+001
 100.00     1.29+001
    .00     1.22+001
COLLFREQ
 500.00     2.00+002
 400.00     3.80+002
 370.00     3.90+002
 350.00     3.80+002
 330.00     3.50+002
 300.00     2.80+002
 250.00     1.05+002
 225.00     3.50+001
 220.00     3.00+001
 210.00     3.30+001
 200.00     4.50+001
 150.00     1.60+003
 120.00     1.00+004
 104.00     3.00+004
    .00     1.82+011
 999.99
 500.00     2.59+001
 395.00     1.03+001
 380.00     9.21+000
 350.00     7.38+000
 300.00     3.81+000
 290.00     3.50+000
 280.00     3.40+000
 275.00     3.56+000
 250.00     4.65+000
 200.00     5.63+000
 170.00     5.86+000
 150.00     5.94+000
 130.00     5.97+000
 100.00     5.94+000
    .00     5.30+000
QUIT
  50 INTEGRATION STEPS, HT = 404.2969
 100 INTEGRATION STEPS, HT = 395.0234
 150 INTEGRATION STEPS, HT = 385.6563
192 INTEGRATION STEPS USED IN WAVFLO
```

SAMPLE OUTPUT

```
SMALLEST INTEGRATION INTERVAL BETWEEN LWSTHT' =   375.0 KM AND WKBHT' =      .0 KM IS     .0489 KM
AVERAGE  INTEGRATION INTERVAL BETWEEN LWSTHT' =   375.0 KM AND WKBHT' =      .0 KM IS     .9665 KM
```

SAMPLE OUTPUT

ELECTRIC DIPOLE TRANSMITTER

	Z		Y		X	
	MAG	ANG	MAG	ANG	MAG	ANG
V	.599227-005	-.783	.255878-009	3.096	.291676-008	-3.138
HB	.713075-005	1.808	.304493-009	-.596	.347092-008	-.547
HE	.724110-005	.213	.309205-009	-2.191	.352463-008	-2.142

MAGNETIC DIPOLE TRANSMITTER

	Z		Y		X	
	MAG	ANG	MAG	ANG	MAG	ANG
V	.717382-005	-2.904	.306331-009	.974	.349188-008	1.023
HB	.213118-003	1.808	.910042-008	-.596	.103736-006	-.548
HE	.213036-003	.238	.909693-008	-2.167	.103696-006	-2.118

V. MODE SUMS

The excitation factors defined in the previous section are useful for calculating the strengths of the electric field components in the earth-ionosphere guide. In this section explicit expressions are given for calculating those field components. The formulas given are asymptotic and restricted to the range

$$|k\rho\sin\theta_N| >> 1 \text{ and } |k(\pi a-\rho)\sin\theta_N| >> 1$$

with k the free space wave number, θ_N the eigenangle for Mode N, a the earth's radius and ρ the transmitter receiver great circle separation. The mode sums are

$$E_j^m\left(\frac{\mu v}{m}\right) = \frac{C_1 I\ell f_{kHz}^{3/2}}{\left(\sin\left(\frac{\rho}{a}\right)\right)^{1/2}} \sum_N \lambda_{jN}^m e^{-ik\rho\sin\theta_N} \qquad \begin{array}{l} j = 1,2,3 \\ m = 1,2,3 \end{array}$$

$$E_j^m\left(\frac{\mu v}{m}\right) = \frac{C_2 IA f_{kHz}^{5/2}}{\left(\sin\left(\frac{\rho}{a}\right)\right)^{1/2}} \sum_N \lambda_{jN}^m e^{-ik\rho\sin\theta_N} \qquad \begin{array}{l} j = 1,2,3 \\ m = 4,5,6 \end{array}$$

where the sum is over the number of modes and

$C_1 = 2.86 \times 10^{-3}$

$C_2 = 5.98 \times 10^{-8}$

f_{kHz} = frequency in kHz

$I\ell$ = electric dipole current moment - Amp-m

IA = magnetic dipole current moment - Amp-m^2

λ_j^m = excitation factors defined in previous section.

The field strengths E_j^m are in terms of microvolts per meter. A program for computing and plotting the mode sums is included in appendix B. The latter takes punched output from the excitation factor program given in appendix A. Results of the mode sum program are shown in figures 1 through 3 for electric dipole (ED) and magnetic dipole (MD) excitation of EX, EY and EZ. The dipole moments in this example have been chosen to correspond to one kilowatt free space radiation.

Fig. 1

HAWAII TO SAN DIEGO GLOBAL NIGHT PROFILE

FREQ=20.000 KHZ TXHT=500.0 KM RXHT=10.0 KM
AZIM=58.5 DEGREES CODIP=39.0 DEGREES
MAGFLD=4.31×10^{-5} W/M^2 SIGMA= 4.64×10^{0} S/M EPSR= 81.0
IL= 1.69×10^{4} A-M IA= 4.03×10^{7} A-M^2

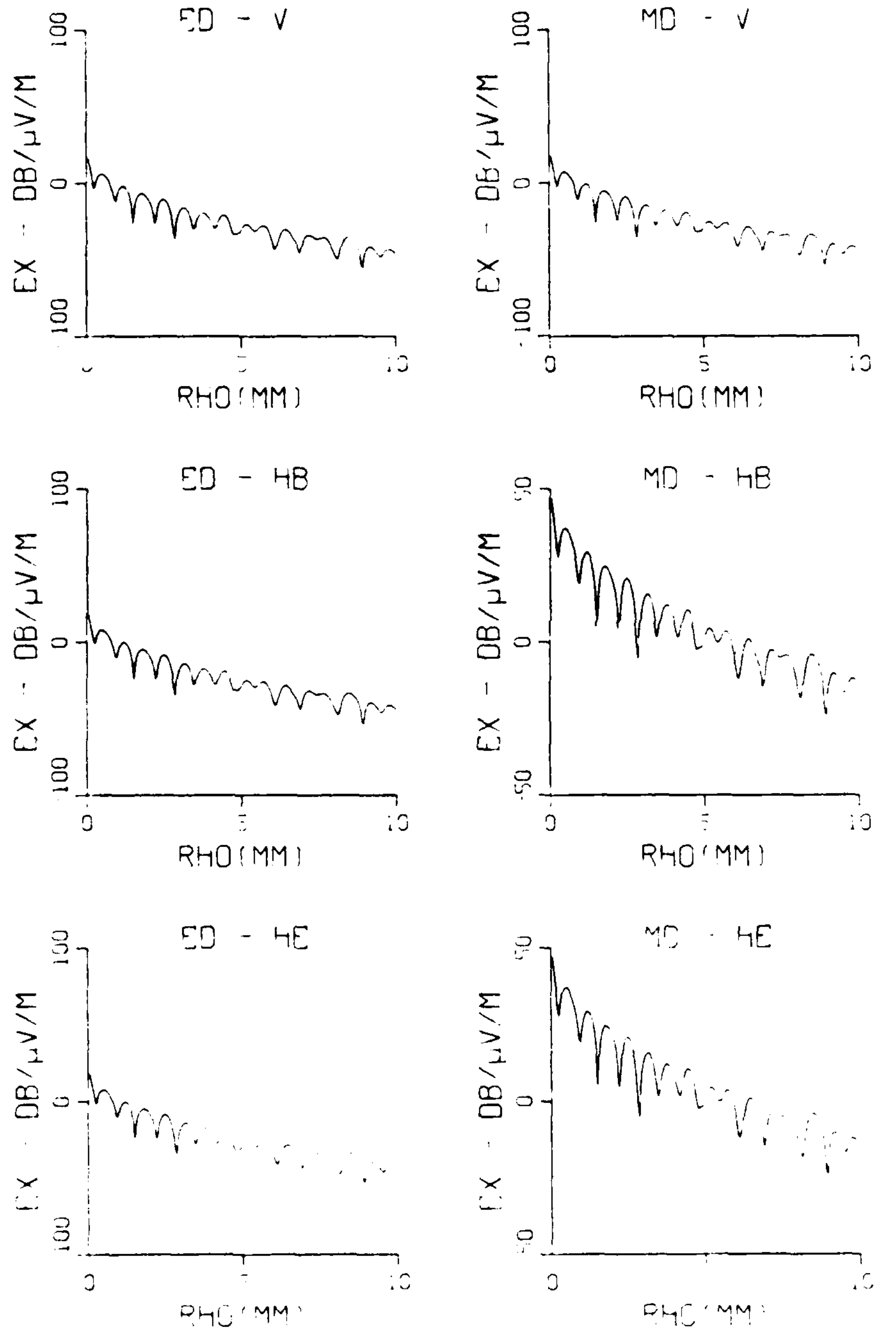


Fig. 2

HAWAII TO SAN DIEGO GLOBAL NIGHT PROFILE

FREQ=20.000 KHZ TXHT=500.0 KM RXHT=10.0 KM
AZIM=58.5 DEGREES CODIP= 39.0 DEGREES
MAGFLD=4.31×10^{-5} W/M^2 SIGMA= 4.64×10^{0} S/M EPSR= 81.0
IL= 1.69×10^{4} A-M IA= 4.03×10^{7} $A\text{-}M^2$

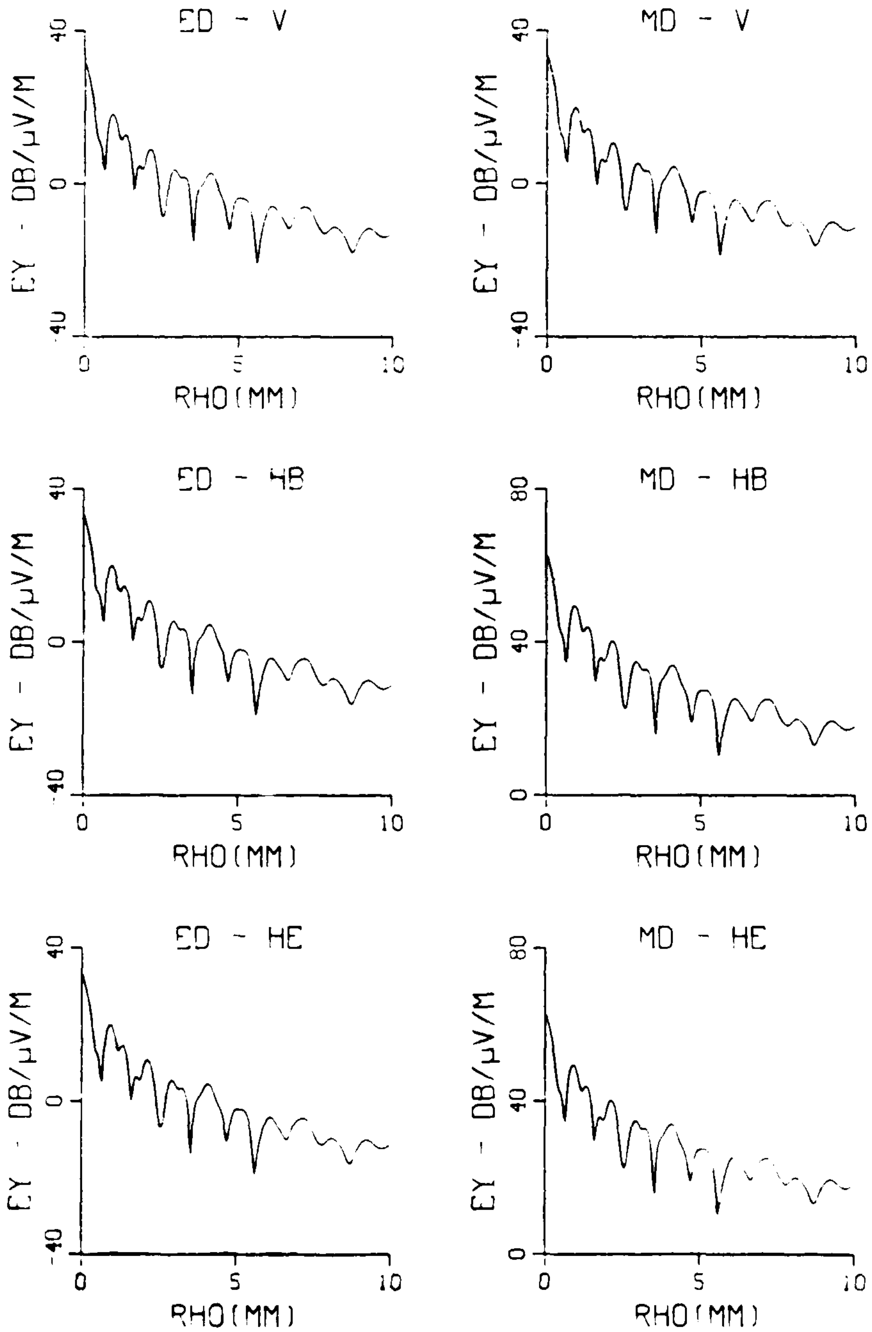
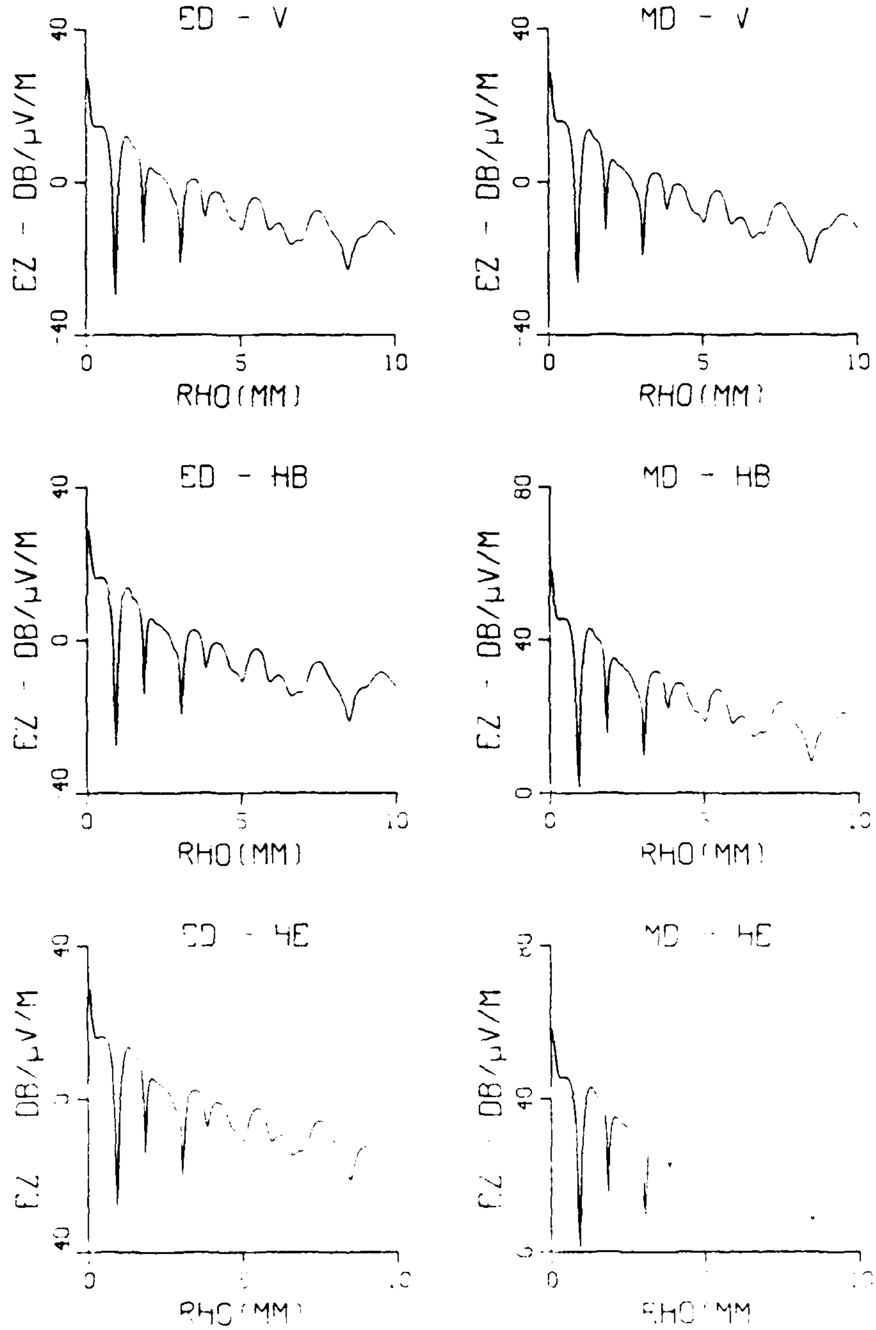


Fig. 3

HAWAII TO SAN DIEGO GLOBAL NIGHT PROFILE

FREQ=20.000 KHZ TXHT=500.0 KM RXHT=10.0 KM
AZIM=58.5 DEGREES CODIP= 39.0 DEGREES
MAGFLD=4.31×10^{-5} W/M^2 SIGMA= 4.64×10^{0} S/M EPSR= 81.0
IL= 1.69×10^{4} A-M IA= 4.03×10^{7} A-M^2



REFERENCES

1. Pappert, RA and LR Hitney, A program to compute ELF/VLF earth-ionosphere modal height gains via WKB methods up to satellite altitudes. Interim Report No. 724 prepared by the Naval Ocean Systems Center for the Defense Nuclear Agency, September 1981.

2. Budden, KG, Radio waves in the ionosphere, Cambridge University Press, Cambridge 1961.

3. Pappert, RA, and LR Shockey, A program for computing earth ionosphere ELF/VLF excitation factors for satellite-borne antennas. Interim Report No. 741 prepared by the Naval Electronics Laboratory Center (now NOSC) for the Defense Atomic Support Agency (now DNA), April 1974.

4. Smith, RR, A program to compute ionospheric height gain functions and field strengths, Interim Report No. 711 prepared by the Naval Electronics Laboratory Center (now NOSC) for the Defense Atomic Support Agency (now DNA), December 1970.

5. Sheddy, CH, RA Pappert, Y Gough and WF Moler, A FORTRAN program for mode constants in an earth-ionosphere waveguide of arbitrary electron density distribution. Interim Report No. 683 prepared by the Naval Electronics Laboratory Center (now NOSC) for the Defense Atomic Support Agency (now DNA), May 1968.

6. Clemmow, PC, and J Heading, Coupled forms of the differential equations governing radio propagation in the ionosphere, Proc. Cambridge Phil. Soc., 50, 1954, pp 319-333.

7. Pappert, RA, Excitation of the earth-ionosphere waveguide by a vertical dipole at satellite heights, Interim Report No. 723 prepared by the Naval Electronics Laboratory Center (now NOSC) for the Defense Atomic Support Agency (now DNA), June 1972.

8. Pappert, RA, Excitation of the earth-ionosphere waveguide by point dipoles at satellite heights. Radio Science 8(6), 1973, pp 535-545.

9. Sheddy, CH, A general analytic solution for reflection from a sharply bounded anisotropic ionosphere, Radio Science, Vol 3, 1968, pp 792-795.

10. Staff of the Computation Laboratory, Tables of the modified Hankel functions of order one third and of their derivatives, Harvard University Press, Cambridge, MA, 1945.

APPENDIX A

SATELLITE EXCITATION FACTOR PROGRAM LISTING

```
C       PROGRAM DRIVER
C     THE DRIVER PROGRAM ALTERNATELY CALLS
C     FOR THE INPUT OF DATA ON THE STANDARD
C     INPUT UNIT AND CALLS FOR THE COMPUTATION
C     OF HEIGHT GAIN FUNCTIONS BY WAVFLD.
C
        IMPLICIT  REAL*8 (A-H,O-Z)
        COMPLEX*16 EX(129,6),EY(129,6),EZ(129,6) ,
      $            HX(129,6),HY(129,6),HZ(129,6)
C
C
        ISTART = 1
        PRINT 905
   100  CALL XINPUT (ISTART, ISTOP)
   150  ISTART=0
        DO 200 J=1,129
        DO 200 K=1,6
        EX(J,K) = 0.0
        EY(J,K) = 0.0
        EZ(J,K) = 0.0
        HX(J,K) = 0.0
        HY(J,K) = 0.0
        HZ(J,K) = 0.0
   200  CONTINUE
        CALL WAVFLD(EX,EY,EZ,HX,HY,HZ)
C
   600  IF (ISTOP.EQ.0)  GO TO 100
C905   FORMAT('1',//)
C
C
        END
```

```
      SUBROUTINE XINPUT (ISTART, ISTOP)
C
C   XINPUT READS IN IONOSPERIC INPUT
C   DATA (VIA NAMELIST), AS WELL AS
C   ELECTRON OR ION DENSITY AND
C   COLLISION FREQUENCY PROFILES. THIS
C   ROUTINE SHOULD BE CALLED PRIOR TO
C   CALLING WAVFLD.  ISTART = 1 INDICATES
C   THAT A NEW SET OF INPUT DATA IS TO BE
C   READ. ISTART = 0 INDICATES AN UPDATE
C   OF EXISTING DATA.  ISTOP = 1 INDICATES
C   THE END OF INPUT DATA FOR THIS JOB.
C   ISTOP = 0 INDICATES MORE INPUT DATA
C   APPEARS IN THE DATA DECK.
C
      IMPLICIT REAL *8 (A-H,O-Z)
      DIMENSION ED(5),COLL(5),TEM(100,5),SAVE(100),RMAG(4),RANG(4)
      COMMON/EXTRA/XTRA,R
      COMMON/STOR/ STORE
      COMMON/WFINPT/THETA,FREQ,AZIM,CODIP,MAGFLD,COEFNU(5),EXPNU(5),
     $             TOPHT,LWSTHT,WKBHT,DELHT,H,ALPHA,SIGMA,EPSLON
      COMMON/WF FLAG/PRECSN,ISO,IDBG
      COMMON/WFPROF/ENHT(100),ENLOG(100,5),COLLHT(25),COLLFR(25,5),
     $             LHT,MHT,CHARGE(5),RATIOM(5),NRSPEC
      COMMON/EXC IN/TXHT,RXHT
      COMMON/DELTA/NUMDIV,STPFLG
      CHARACTER*8 BCD(10)
      INTEGER STPFLG
      REAL*8  MAGFLD, LWSTHT
      COMPLEX*16 R(4),XTRA,I/(0.0D0,1.0D0)/,THETA
C
      NAMELIST/DATUM/THETA,FREQ,IDBG,TOPHT,LWSTHT,WKBHT,DELHT,PRECSN,
     $ AZIM,CODIP,MAGFLD,COEFNU,EXPNU,ALPHA,SIGMA,EPSLON,EPSLNO,
     $       TXHT,RXHT,  H,RMAG,RANG,XTRMAG,XTRANG,NUMDIV,STPFLG,
     $ NRSPEC,CHARGE,RATIOM
C
      DATA DEGRAD/1.745329250-2/
C
C
      IDAT = 0
      ISTOP = 1
      ICOLL = 1
      IF (ISTART.NE.0)  ICOLL = 0
100   READ 900, BCD
      PRINT 901, BCD
      IF (BCD(1) .EQ. 'DATUMFOL') GO TO 200
      IF (BCD(1) .EQ. 'COLLFREQ') GO TO 400
      IF (BCD(1) .EQ. 'PROFILE ') GO TO 500
      IF (BCD(1) .EQ. 'STOP    ') GO TO 120
      IF (BCD(1) .NE. 'QUIT    ') GO TO 800
      ISTOP = 0
120   IF (IDAT.EQ.0.AND.ISTART.NE.0)  GO TO 800
      IF (NRSPEC.EQ.0)  NRSPEC = 1
      IF (ICOLL.NE.0)  RETURN
```

```
      COLLHT(1) = STORE
      COLLHT(2) = 0.0
      DO 150  J = 1,NRSPEC
      COLLFR(2,J) = DLOG(COEFNU(J)*DEXP(STORE*1000.*EXPNU(J)))
      COLLFR(1,J) = COLLFR(2,J)-1000.0*STORE*EXPNU(J)
150   CONTINUE
      RETURN
C
200   IDAT = 1
      PREVNU = COEFNU(1)
      COEFNU(1) = 0.0
      READ (5, DATUM)
      IF (COEFNU(1).NE.0.0)  ICOLL = 0
      IF (COEFNU(1).EQ.0.0)  COEFNU(1) = PREVNU
      IF (NUMDIV.NE.0  .OR. STPFLG.EQ.1) GO TO 249
      NUM=DLOG( 75. + DSQRT(DELHT * DSQRT(FREQ) ) )
      NUMDIV=2 **NUM
249   WRITE (6, DATUM)
      DO 20 JK=1,4
      RANG(JK) = RANG(JK)*DEGRAD
20    R(JK) = RMAG(JK)*(DCOS(RANG(JK))+I*DSIN(RANG(JK)))
      XTRA = XTRMAG*(DCOS(XTRANG)+I*DSIN(XTRANG))
      PUNCH 920
      PUNCH 921,THETA
      PUNCH 922,FREQ
      PUNCH 923,TXHT,RXHT
      EPSR = EPSLON/EPSLNO
      PUNCH 925,AZIM,CODIP,MAGFLD,SIGMA,EPSR
      PUNCH 924
      GO TO 100
C
C
400   ICOLL = 1
      L = 1
410   READ 902, HT, COLL
      IF (DABS(HT-999.99).LT.0.001) GO TO 430
      IF (L.GT.25)  GO TO 800
      IF(L .NE. 1 .AND. HT .GE. COLLHT(L-1)) GO TO 800
      DO 411 M=1,NRSPEC
      IF(COLL(M) .LE. 0.0) COLL(M)=1.0D-40
411   CONTINUE
      PRINT 903,HT,(COLL(M),M=1,NRSPEC)
      COLLHT(L) = HT
      DO 415 M=1,NRSPEC
415   COLLFR(L,M)=DLOG(COLL(M))
      L = L + 1
      GO TO 410
430   PRINT 903,HT
      LL = L-1
      DO 420 J=1,LL
      DO 420 M=1,NRSPEC
      JJ= LL+1-J
420   TEM(JJ,M) = COLLFR(J,M)
      DO 425 J=1,LL
      DO 425 M=1,NRSPEC
425   COLLFR(J,M) = TEM(J,M)
      LL = L-1
```

```
      DO 440 J=1,LL
      COLLHT(J) = -COLLHT(J)+STORE
      JJ = LL+1-J
440   SAVE(JJ) = COLLHT(J)
      DO 450 J=1,LL
      COLLHT(J) = SAVE(J)
450   PRINT 903, COLLHT(J),(COLLFR(J,M),M=1,NRSPEC)
      GO TO 100
C
500   READ 900,BCD
      PRINT 901,BCD
      PUNCH 900,BCD
      L=1
510   READ 902,HT,ED
      IF(DABS(HT-999.99) .LT. 0.001) GO TO 530
      IF (L.GT.100) GO TO 800
      IF(L .NE. 1 .AND. HT .GE. ENHT(L-1)) GO TO 800
      ENHT(L) = HT
      IF(NRSPEC .EQ. 3) ED(3) = ED(2)-ED(1)
      DO 511 M=1,NRSPEC
      IF(ED(M) .LE. 0.0) ED(M)=1.0D-40
511   CONTINUE
      PRINT 903,HT,(ED(M),M=1,NRSPEC)
      DO 515 M=1,NRSPEC
515   ENLOG(L,M) = DLOG(ED(M))
      L = L+1
      GO TO 510
530   PRINT 903,HT
      LL = L-1
      DO 520 J=1,LL
      DO 520 M=1,NRSPEC
      JJ = LL+1-J
520   TEM(JJ,M) = ENLOG(J,M)
      DO 525 J=1,LL
      DO 525 M=1,NRSPEC
525   ENLOG(J,M) = TEM(J,M)
      TXHT = -TXHT+ENHT(1)
      RXHT = -RXHT+ENHT(1)
      TOPHT = -TOPHT+ENHT(1)
      LWSTHT = -LWSTHT+ENHT(1)
      WKBHT = -WKBHT+ENHT(1)
      TEMP = LWSTHT
      LWSTHT = TOPHT
      TOPHT = TEMP
      H =-H +ENHT(1)
      LL = L-1
      STORE = ENHT(1)
      DO 540 J=1,LL
      ENHT(J) = -ENHT(J)+STORE
      JJ = LL+1-J
540   SAVE(JJ) = ENHT(J)
      DO 550 J=1,LL
      ENHT(J) = SAVE(J)
550   PRINT 903, ENHT(J),(ENLOG(J,M),M=1,NRSPEC)
      GO TO 100
C
C     ERROR EXIT
```

```
 800  PRINT 910
      STOP
 900  FORMAT (10A8)
 901  FORMAT (1X,10A8)
 902  FORMAT(F7.2,4X,5(1X,E9.2))
 903  FORMAT(1X,F7.2,5X,5(1PE9.2,6X))
 910  FORMAT ('1ERROR IN DATA DECK DETECTED',/,
     $ ' IN SUBROUTINE XINPUT')
 920  FORMAT(' &DATUM')
 921  FORMAT(' THETA=(',F6.3,',',F7.3,'),')
 922  FORMAT(' FREQ=',F7.3,',')
 923  FORMAT(' TXHT=',F8.3,', RXHT=',F8.3,',')
 924  FORMAT(' &END')
 925  FORMAT(' AZIM=',F7.3,',','CODIP=',F7.3,',','MAGFLD=',D11.4,',',
     $  'SIGMA=',D10.3,',','EPSR=',F4.1,',')
      END
```

```
      SUBROUTINE WAVFLD(EX,EY,EZ,HX,HY,HZ)
C
C  WAVFLD CALLS FOR THE DOWNWARD
C  INTEGRATION, AND THEN PERFORMS THE
C  BACK SUBSTITUTION OF NORMALIZING
C  VALUES (SAVED AS DATA BY WFSTOR).
C  FIELD STRENGTHS ARE COMPUTED AT
C  HEIGHTS FROM TOPHT TO LWSTHT AT
C  DELHT INCREMENTS, AND ARE RETURNED
C  IN THE LISTS EX, EY, EZ, HX, HY, HZ.
C
      IMPLICIT REAL *8 (A-H,O-Z)
      COMMON/EXTRA/XTRA,R(4)
      COMMON/STOR/ STORE
      COMMON/WF FLAG/PRECSN,ISO,IDBG
      COMMON/WF SAVE/P(4,2),M31,M32,M33,ORTHO,ANORM,BNORM,HT,LEVL
      COMMON/CS/C,S,CI,SI
      COMMON/WFINPT/THETA,FREQ,AZMUTH,CODIP,MAGFLD,CEFFNU(5),EXPNU(5),
     $ TOPHT,LWSTHT,WKBHT,DELHT,H,ALPHA,SIGMA,EPSLON
      COMMON/EXC IN/TXHT,RXHT
      COMMON/D RX TX/EY D,EZ D,EX TXHT,EY TXHT,EZ TXHT,
     $            EX RXHT,EY RXHT,EZ RXHT,
     $            HX RXHT,HY RXHT,HZ RXHT
      REAL*3  MAGFLD, LWSTHT
      COMPLEX*16 I/(0.0D0,1.0D0)/
      COMPLEX*16 EX(129,6),EY(129,6),EZ(129,6),
     $             HX(129,6),HY(129,6),HZ(129,6),
     $             XTRA,R,P,M31,M32,M33,ORTHO,C,S,CI,SI,THETA,EYD,EZD,
     $             EXTXHT,EYTXHT,EZTXHT,EXRXHT,EYRXHT,EZRXHT,HXRXHT,
     $             HYRXHT,HZRXHT,B(12),W(4,12),OSUM,RBAR11,RBAR22,EXT,EYT,
     $             EZT,HXT,HYT,HZT,HYD,HYENHT,HYREC,EXC,EFIELD(3,3),
     $             HFIELD(3,3),EYDEE,EYENHT,EXENHT,EYREC,EXREC
      DIMENSION EMAG(3,3),EANG(3,3),HMAG(3,3),HANG(3,3)
      DIMENSION EXMAG(6),EYMAG(6),EZMAG(6),HXMAG(6),HYMAG(6),HZMAG(6),
     $            EXANG(6),EYANG(6),EZANG(6),HXANG(6),HYANG(6),HZANG(6)
      EQUIVALENCE(TOPHT,D)
      DATA PI/3.141592653D0/
C
C
      JHT = TOPHT/DELHT+1.01
      TEST = (JHT-1)*DELHT-TOPHT
      IF(DABS(TEST) .GT. 0.0001) GO TO 800
      MHT = LWSTHT/DELHT+1.01
      TEST = (MHT-1)*DELHT-LWSTHT
      IF(DABS(TEST) .GT. 0.0001) GO TO 800
C
      CALL WFINTG(TOPHT,LWSTHT,DELHT,0)
      CALL DECOMP(LWSTHT)
      CALL WKB
C
C  COMBINE SOLUTIONS AT GROUND SO THAT
C  THEY SATISFY BOUNDARY CONDITION.
      CALL WF BNDY(E)
C
```

```
C   PERFORM BACK SUBSTITUTION OF
C   NORMALIZING VALUES.
      O SUM = 0.0
      PRODA = 1.0
      PRODB = 1.0
      IHT=MHT
      CALL WFLOAD
      GO TO 25
C
   21 O SUM = O SUM*ANORM/BNORM+ORTHO
      PRODA = PRODA * ANORM
      IF(PRODA .LT. 1.0D-30) PRODA = 0.0
      PRODB = PRODB * BNORM
      CALL WF LOAD
      DO 23 J=1,4
      P(J,2) = (P(J,2)-O SUM*P(J,1))*PRODB
   23 P(J,1) = P(J, 1) * PRODA
C
C   COMPUTE FIELD STRENGTHS
C   AT PROFILE HEIGHTS.
   25 DO 26 N=1,11,2
      DO 26 J=1,4
26    W(J,N) = P(J,1)*B(N)+P(J,2)*B(N+1)
      DO 27 N=1,11,2
      NN = (N+1)/2
      EX(IHT,NN) = W(1,N)
      EY(IHT,NN) = -W(2,N)
      EZ(IHT,NN) = -(S*W(4,N)+M31*W(1,N)-M32*W(2,N))/(1.+M33)
      HX(IHT,NN) = W(3,N)
      HY(IHT,NN) = W(4,N)
      HZ(IHT,NN) = -S*W(2,N)
      EXMAG(NN) = CDABS(EX(IHT,NN))
      EYMAG(NN) = CDABS(EY(IHT,NN))
      EZMAG(NN) = CDABS(EZ(IHT,NN))
      HXMAG(NN) = CDABS(HX(IHT,NN))
      HYMAG(NN) = CDABS(HY(IHT,NN))
      HZMAG(NN) = CDABS(HZ(IHT,NN))
      EXANG(NN) = CDANG(EX(IHT,NN))
      EYANG(NN) = CDANG(EY(IHT,NN))
      EZANG(NN) = CDANG(EZ(IHT,NN))
      HXANG(NN) = CDANG(HX(IHT,NN))
      HYANG(NN) = CDANG(HY(IHT,NN))
      HZANG(NN) = CDANG(HZ(IHT,NN))
   27 CONTINUE
C
      IHT = IHT + 1
      IF (IHT.LE.JHT)  GO TO 21
      IF(LEVL .NE. 0) PRINT 903,LEVL
      CALL RBARS(C,S,RBAR11,RBAR22)
      CALL HTGAIN(D,EXT,EYT,EZT,HXT,HYT,HZT)
      HYD = HYT
      EYDEE = EYT
      CALL HTGAIN(STORE,EXT,EYT,EZT,HXT,HYT,HZT)
      HYENHT = HYT/HYD
      EYENHT = EYT/EYDEE
      EXENHT = EXT/HYD
      IF(RXHT .EQ. STORE) GO TO 50
```

```
      CALL HTGAIN(RXHT,EXT,EYT,EZT,HXT,HYT,HZT)
      HYREC = HYT/HYD
      EYREC = EYT/EYDEE
      EXREC = EXT/HYD
      GO TO 60
   50 HYREC = HYENHT
      EYREC = EYENHT
      EXREC = EXENHT
   60 EXC = RBAR11*XTRA/(CDSQRT(S)*S**2* (1.+RBAR11)**2*(1.-R(2)*
     $  RBAR22)*HYENHT**2)
      EFIELD(1,1) = C*CDSQRT(S)*EXC*CDEXP(-.25*PI*I)*HYREC*((1.-R(2)*
     $ RBAR22)*HY(IHT-1,1)+R(3)*RBAR22*EY(IHT-1,1))*2.0*(1.+RBAR11)
      EFIELD(1,2) = C*EXC*CDEXP(.75*PI*I)*EYREC*(R(4)*RBAR11*HY(IHT-1,1)
     $ +(1.-R(1)*RBAR11)*EY(IHT-1,1))*2.0*(1.+RBAR22)
     $  /CDSQRT(S)
      EFIELD(1,3) = C*EXC*CDEXP(1.75*PI*I)*EXREC*((1.-R(2)*RBAR22)*
     $ HY(IHT-1,1)+R(3)*RBAR22*EY(IHT-1,1))*2.0*(1.+RBAR11)
     $  /CDSQRT(S)
       EFIELD(2,1) = C*CDSQRT(S)*EXC*CDEXP(-.25*PI*I)*HYREC*((1.-R(2)
     $ *RBAR22)*HY(IHT-1,2)+R(3)*RBAR22*EY(IHT-1,2))*2.0*(1.+RBAR11)
      EFIELD(2,2) = C*EXC*CDEXP(.75*PI*I)*EYREC*(R(4)*RBAR11*HY(IHT-1,2)
     $ +(1.-R(1)*RBAR11)*EY(IHT-1,2))*2.0*(1.+RBAR22)
     $  /CDSQRT(S)
      EFIELD(2,3) = C*EXC*CDEXP(1.75*PI*I)*EXREC*((1.-R(2)*RBAR22)*
     $ HY(IHT-1,2)+R(3)*RBAR22*EY(IHT-1,2))*2.0*(1.+RBAR11)
     $  /CDSQRT(S)
      EFIELD(3,1) = C*CDSQRT(S)*EXC*CDEXP(-.25*PI*I)*HYREC*((1.-R(2)*
     $ RBAR22)*HY(IHT-1,3)+R(3)*RBAR22*EY(IHT-1,3))*2.0*(1.+RBAR11)
      EFIELD(3,2) = C*EXC*CDEXP(.75*PI*I)*EYREC*(R(4)*RBAR11*HY(IHT-1,3)
     $ +(1.-R(1)*RBAR11)*EY(IHT-1,3))*2.0*(1.+RBAR22)
     $  /CDSQRT(S)
      EFIELD(3,3) = C*EXC*CDEXP(1.75*PI*I)*EXREC*((1.-R(2)*RBAR22)*
     $ HY(IHT-1,3)+R(3)*RBAR22*EY(IHT-1,3))  *2.0*(1.+RBAR11)
     $  /CDSQRT(S)
      HFIELD(1,1) = C*CDSQRT(S)*EXC*CDEXP(-.75*PI*I)*HYREC*((1.-R(2)*
     $ RBAR22)*HY(IHT-1,4)+R(3)*RBAR22*EY(IHT-1,4))*2.0*(1.+RBAR11)
      HFIELD(1,2) = C*EXC*CDEXP(.25*PI*I)*EYREC*(R(4)*RBAR11*HY(IHT-1,4)
     $ +(1.-R(1)*RBAR11)*EY(IHT-1,4))*2.0*(1.+RBAR22)
     $  /CDSQRT(S)
      HFIELD(1,3) = C*EXC*CDEXP(1.25*PI*I)*EXREC*((1.-R(2)*RBAR22)*
     $ HY(IHT-1,4)+R(3)*RBAR22*EY(IHT-1,4))*2.0*(1.+RBAR11)
     $  /CDSQRT(S)
      HFIELD(2,1)  = C*CDSQRT(S)*EXC*CDEXP(-.75*PI*I)*HYREC*((1.-R(2)
     $ *RBAR22)*HY(IHT-1,5)+R(3)*RBAR22*EY(IHT-1,5))*2.0*(1.+RBAR11)
      HFIELD(2,2) = C*EXC*CDEXP(.25*PI*I)*EYREC*(R(4)*RBAR11*HY(IHT-1,5)
     $ +(1.-R(1)*RBAR11)*EY(IHT-1,5))*2.0*(1.+RBAR22)
     $  /CDSQRT(S)
      HFIELD(2,3) = C*EXC*CDEXP(1.25*PI*I)*EXREC*((1.-R(2)*RBAR22)*
     $ HY(IHT-1,5)+R(3)*RBAR22*EY(IHT-1,5))*2.0*(1.+RBAR11)
     $  /CDSQRT(S)
      HFIELD(3,1) = C*CDSQRT(S)*EXC*CDEXP(-.75*PI*I)*HYREC*((1.-R(2)*
     $ RBAR22)*HY(IHT-1,6)+R(3)*RBAR22*EY(IHT-1,6))*2.0*(1.+RBAR11)
      HFIELD(3,2) = C*EXC*CDEXP(.25*PI*I)*EYREC*(R(4)*RBAR11*HY(IHT-1,6)
     $ +(1.-R(1)*RBAR11)*EY(IHT-1,6))*2.0*(1.+RBAR22)
     $  /CDSQRT(S)
      HFIELD(3,3) = C*EXC*CDEXP(1.25*PI*I)*EXREC*((1.-R(2)*RBAR22)*
     $ HY(IHT-1,6)+R(3)*RBAR22*EY(IHT-1,6))  *2.0*(1.+RBAR11)
```

```
     $   , CDSQRT(S)
       DO 100 J=1,3
       DO 100 K=1,3
       EMAG(J,K) = CDABS(EFIELD(J,K))
       EANG(J,K) = CDANG(EFIELD(J,K))
       HMAG(J,K) = CDABS(HFIELD(J,K))
       HANG(J,K) = CDANG(HFIELD(J,K))
 100   CONTINUE
       PRINT 920
       PRINT 921
       PRINT 922
       PRINT 923,(EMAG(1,K),EANG(1,K),K=1,3)
       PRINT 924,(EMAG(2,K),EANG(2,K),K=1,3)
       PRINT 925,(EMAG(3,K),EANG(3,K),K=1,3)
       PUNCH 927,(EMAG(1,K),EANG(1,K),K=1,3)
       PUNCH 928,(EMAG(2,K),EANG(2,K),K=1,3)
       PUNCH 929,(EMAG(3,K),EANG(3,K),K=1,3)
       PRINT 930
       PRINT 921
       PRINT 922
       PRINT 923,(HMAG(1,K),HANG(1,K),K=1,3)
       PRINT 924,(HMAG(2,K),HANG(2,K),K=1,3)
       PRINT 925,(HMAG(3,K),HANG(3,K),K=1,3)
       PUNCH 927,(HMAG(1,K),HANG(1,K),K=1,3)
       PUNCH 928,(HMAG(2,K),HANG(2,K),K=1,3)
       PUNCH 929,(HMAG(3,K),HANG(3,K),K=1,3)
       PRINT 926
       RETURN
C
C
   800 PRINT 902
       STOP
C
 902   FORMAT (' ERROR IN WAVFLD',/,
     $ ' DELHT DOES NOT DIVIDE TOPHT-LWSTHT EVENLY')
 903   FORMAT(' ','LEVL NOT ZERO. LEVL = ',I3)
 920   FORMAT('1',14X,'ELECTRIC DIPOLE TRANSMITTER'//)
 921   FORMAT(' ',31X,'Z',25X,'Y',25X,'X'/)
 922   FORMAT(' ',24X,3('MAG',9X,'ANG',11X))
 923   FORMAT(' ',14X,'V',4X,3(E13.6,2X,F6.3,5X)/)
 924   FORMAT(' ',13X,'HB',4X,3(E13.6,2X,F6.3,5X)/)
 925   FORMAT(' ',13X,'HE',4X,3(E13.6,2X,F6.3,5X)/)
 926   FORMAT('1')
 927   FORMAT('  V',2X,3(E13.6,2X,F6.3,4X))
 928   FORMAT(' HB',2X,3(E13.6,2X,F6.3,4X))
 929   FORMAT(' HE',2X,3(E13.6,2X,F6.3,4X))
 930   FORMAT('0',14X,'MAGNETIC DIPOLE TRANSMITTER'//)
       END
```

```
      SUBROUTINE WF INTG(TOPHT,LWSTHT,DELHT,IFLAG)
C
C  WF INTG PERFORMS THE INTECRATION OF THE P MATRIX THROUGH THE
C  IONOSPHERE,USING THE TECHNIQUES GIVEN BY PITTEWAY.  ACCURACY IS
C  MAINTAINED BY ADJUSTING THE STEPSIZE SO THAT THE P MATRIX IS
C  COMPUTED WITH SUFFICIENT ACCURACY.
C
C IFLAG=0  INTEG FOR THETA ONLY
C IFLAG=1  INTEG FOR THETA AND THETA-DTHETA
C
      IMPLICIT REAL *8 (A-H,O-Z)
      COMMON/WF FLAG/PRECSN,ISO,IDBG
      COMMON/P MIX/P(8),PI(8)
      COMMON/WF SAVE/P SAVE(8),M31 SAV,M32 SAV,M33 SAV,
     $          ORTHO,ANORM,BNORM,HT,LEVL
      COMMON/M MTX/M(3,3)
      COMMON/WFPROF/ENHT(100),ENLOG(100,5),COLLHT(25),COLLFR(25,5),
     $               LHT,MHT,CHARGE(5),RATIOM(5),NRSPEC
      INTEGER SVFLAG
      REAL*8  LWSTHT
      COMPLEX*16 M31 SAV,M32 SAV,M33 SAV,ORTHO,M
      COMPLEX*16 P,PI,P SAVE,PREV P,TEMP P,DPDH,PV DPDH,DPIDH
      DIMENSION PREVP(8),TEMPP(8),DPDH(8),PV DPDH(8),DPIDH(8)
      DIMENSION PR(16),TPR(16)
      EQUIVALENCE (P,PR),(TEMPP,TPR)
      DATA  EPSHT /5.0D-4/
      DATA  DHMIN /1.0D-3/
C  MINIMUM STEP-SIZE ALLOWED
C
C
      CALL INIT T
      CALL T MTRX(TOP HT)
      CALL WF INIT(P)
      CALL P DERIV(P,DPDH)
      IF(IFLAG .EQ. 0) GO TO 11
      CALL TI MTRX
      CALL WF INIT(PI)
      CALL P DERIV(PI,DPIDH)
   11 CONTINUE
C
      ISTEPS = 0
      KMAX = 0
      LEVL = 0
      HT = TOPHT
      CALL XFER(P,P SAVE,8)
      M31 SAV = M(3,1)
      M32 SAV = M(3,2)
      M33 SAV = M(3,3)
      CALL WF STOR
      WFHT = TOPHT - DELHT
      DELH2 = 0.125D0*DELHT
      SVFLAG=0
C
C  DETERMINE NEXT STEPSIZE TO USE.
```

```
   10    IF (SVFLAG.EQ.1) DELH2=SAVDH2
         SVFLAG=0
         NODBL = 0
         HTO=HT
         CALL XFER(P,PREVP,8)
         CALL XFER(DPDH,PV DPDH,8)
         HTLIM = WFHT
         IF (ENHT(LHT+1).GT.HTLIM+EPSHT)
        $ HTLIM = ENHT(LHT+1)
         IF (COLLHT(MHT+1).GT.HTLIM+EPSHT)
        $ HTLIM = COLLHT(MHT+1)
         IF (HTO-DELH2.GE.HTLIM+EPSHT)  GO TO 50
         SAVDH2 = DELH2
         SVFLAG = 1
         DELH2 = HTO - HTLIM
C
C    PERFORM NEXT INTEGRATION STEP.
   50 CALL WF STEP(P,DPDH,HT,DELH2,0)
         CALL XFER(P,TEMPP,8)
         M31 SAV = M(3,1)
         M32 SAV = M(3,2)
         M33 SAV = M(3,3)
         HT=HTO
         CALL XFER(PREVP,P,8)
         CALL XFER(PV DPDH,DPDH,8)
         DELH=0.5*DELH2
         CALL WF STEP(P,DPDH,HT,DELH,1)
         CALL P DERIV(P,DPDH)
         CALL WF STEP(P,DPDH,HT,DELH,2)
C    CHECK ACCURACY OF RESULT.
         PMAX = 0.0
         DO 85 J=1,16
         PABS = DABS(PR(J)-TPR(J))
         IF(PMAX .LT. PABS) PMAX = PABS
   85 CONTINUE
C    ADJUST STEPSIZE IF NECESSARY.
         IF (PMAX.LT.PRECSN)  GO TO 100
C
         IF (DELH.GT.DHMIN)  GO TO 95
         IF (KMAX.EQ.0)  PRINT 900, HT
         KMAX = 1
         GO TO 100
   95 CONTINUE
         DELH2 = 0.5 * DELH2
         NODBL = 1
         IF (PMAX.LT.10.0*PRECSN)  GO TO 99
         DELH2 = 0.25 * DELH2
         NODBL = 0
   99 CONTINUE
         HT=HTO
         CALL XFER(PREVP,P,8)
         CALL XFER(PV DPDH,DPDH,8)
         SVFLAG=0
         GO TO 50
C
  100 CALL WF SCAL(P,0)
         CALL XFER(P,P SAVE,3)
```

```
      IF(HT .LT. WFHT+EPSHT) CALL WF STOR
      CALL P DERIV(P,DPDH)
      IF(IFLAG .EQ. 0) GO TO 72
      HT = HI0
      CALL WF STEP(PI,DPIDH,HT,DELH,3)
      CALL P DERIV(PI,DPIDH)
      CALL WF STEP(PI,DPIDH,HT,DELH,4)
      CALL WF SCAL(PI,1)
      CALL P DERIV(PI,DPIDH)
   72 CONTINUE
C
      ISTEPS = ISTEPS+1
      IF(IDBG .EQ. 0) GO TO 73
      IDIV = ISTEPS/50
      IF(ISTEPS .EQ. 50*IDIV) PRINT 902,ISTEPS,HT
   73 CONTINUE
      IF(NO DBL .EQ. 0 .AND. PMAX .LT. 0.1*PRECSN) DELH2 = 2.0*DELH2
C
C    CHECK INTEGRATION AND PROFILE HEIGHTS.
      IF (HT.LT.LWSTHT+EPSHT)  GO TO 80
      IF (HT.LT.WFHT+EPSHT)  WFHT = WFHT - DELHT
      IF (HT.LT.ENHT(LHT+1)+EPSHT)  LHT = LHT + 1
      IF (HT.LT.COLLHT(MHT+1)+EPSHT)  MHT = MHT + 1
      GO TO 10
C
   80 PRINT 901,ISTEPS
      RETURN
C
  900 FORMAT (' MINIMUM STEPSIZE USED AT HT =',D14.5)
  901 FORMAT (1X,I3,' INTEGRATION STEPS USED IN WAVFLD',/)
  902 FORMAT (1X,I4,' INTEGRATION STEPS, HT =',F9.4)
      END
```

```
      SUBROUTINE T MTRX(HT)
C
C  T MTRX   COMPUTES THE MATRICES
C     M- THE SUSCEPTIBILITY TENSOR
C     T- THE COEFFICIENT MATRIX OF
C        THE LINEAR SYSTEM OF O.D.E.
C          DP/DZ = -IK*T*P.
C   NOTE THAT ON CALL TO  ENTRY INIT T, VARIOUS IONOSPHERIC CONSTANTS
C   ARE COMPUTED.
C
      IMPLICIT REAL *8 (A-H,O-Z)
      COMMON/WF FLAG/PRECSN,ISO,IDBG
      COMMON/M MTX/M11,M21,M31,M12,M22,M32,M13,M23,M33
      COMMON/T MTX/ T11,T31,T41,T12,T32,T42,T14,T34,T44
      COMMON/TM MTX/TM11,TM31,TM41,TM12,TM32,TM42,TM14,TM34,TM44
      COMMON/CS/C,S,CI,SI
      COMMON/WFINPT/THETA,FREQ,AZMUTH,CODIP,MAGFLD,CEFFNU(5),EXPNU(5),
     $ TOPHT,LWSTHT,WKBHT,DELHT,H,ALPHA,SIGMA,EPSLON
      COMMON/WF CON/OMEGA,WAVE NR
      COMMON/WFPROF/ENHT(100),ENLOG(100,5),COLLHT(25),COLLFR(25,5),
     $              LHT,MHT,CHARGE(5),RATIOM(5),NRSPEC
      REAL*8  MAGFLD, LWSTHT,
     $         LSQYSQ, MSQYSQ, NSQYSQ,
     $ LMYSQ, LNYSQ, MNYSQ, NU,
     $           LY,MY,NY
      COMPLEX*16 M(3,3),
     $           M11,M21,M31,M12,M22,M32,M13,M23,M33,
     $           T11,T31,T41,T12,T32,T42,T14,T34,T44,
     $           TM11,TM31,TM41,TM12,TM32,TM42,TM14,TM34,TM44,
     $           C,S,CI,SI,CSQ,SSQ,CSQI,SSQI,
     $           THETA,DTHETA,
     $           D,M13D,M23D,
     $           U,USQ,DD,I,IUD,TA,TB
      DIMENSION  Y(5), YSQ(5),  LY(5),  MY(5),
     $  NY(5), LMYSQ(5), LNYSQ(5), MNYSQ(5), EN(5), NU(5),
     $ LSQYSQ(5), MSQYSQ(5), NSQYSQ(5),
     $           COEF EN(5)
C
      EQUIVALENCE(M11,M)
      DATA PI/3.141592653D0/
      DATA DIR/1.745329252D-2/
      DATA COEFFX/3.182357D03/,COEFFY/1.758796D11/
      DATA I/(0.0D0,1.0D0)/
      DATA VELLT/2.997928D05/
      DATA DTHETA/(5.0D-2,1.0D-2)/
C
C
C   CALCULATE THE MATRIX M.
      M(1,1) = 0.0
      M(1,2) = 0.0
      M(1,3) = 0.0
      M(2,1) = 0.0
      M(2,2) = 0.0
      M(2,3) = 0.0
```

```
      M(3,1) = 0.0
      M(3,2) = 0.0
      M(3,3) = 0.0
C
      CALL WF DENS (HT, EN, NU)
      NFLAG = 0
      DO 20 K=1,NRSPEC
C   ADD IN THE CONTRIBUTIONS TO THE
C   SUSCEPTIBILITY TENSOR M FOR EACH
C   SPECIE IN THE IONOSPHERE.
      IF(EN(K) .LT. 1.0E-3) GO TO 20
      NFLAG = 1
      X = COEF EN(K)*EN(K)
      Z = NU(K)*OV OMGA
      U=1.0-I*Z
      USQ=U*U
      DD = -X / (U * (USQ - YSQ(K)) )
      IUD = (Z+I)*DD
      TA = USQ * DD
      M(1,1) = M(1,1) + TA
      M(2,2) = M(2,2) + TA
      M(3,3) = M(3,3) + TA
      M(2,2) = M(2,2) - MSQYSQ(K) * DD
      TA = MY(K)*IUD
      TB = LNYSQ(K) * DD
      M(1,3) = M(1,3) + TA - TB
      M(3,1) = M(3,1) - TA - TB
      IF (ISO.NE.0)  GO TO 20
      M(1,1) = M(1,1) - LSQYSQ(K) * DD
      M(3,3) = M(3,3) - NSQYSQ(K) * DD
      TA = NY(K)*IUD
      TB = LMYSQ(K) * DD
      M(2,1) = M(2,1) + TA - TB
      M(1,2) = M(1,2) - TA - TB
      TA = LY(K)*IUD
      TB = MNYSQ(K) * DD
      M(3,2) = M(3,2) + TA - TB
      M(2,3) = M(2,3) - TA - TB
   20 CONTINUE
C
      CRVTRM=ALPHA*(H-HT)
      M(1,1) = M(1,1) + CRVTRM
      M(2,2) = M(2,2) + CRVTRM
      M(3,3) = M(3,3) + CRVTRM
C
C   CALCULATE THE MATRIX T.
      D = 1.0/(1.0+M33)
      TM41 = 1.0+M11
      TM32 = M22
      TM14 = D
      IF(NFLAG .EQ. 0) GO TO 40
      M13D = M13*D
      M23D = M23*D
      TM41 = TM41-M31*M13D
      TM11 = M31*D
      TM44 = M13D
      IF(ISO .NE. 0) GO TO 40
```

```
      TM32 = TM32-M32*M23D
      TM31 = M31*M23D-M21
      TM12 = M32*D
      TM42 = M32*M13D-M12
      TM34 = M23D
C
   40 T41 = TM41
      T32 = CSQ+TM32
      T14 = 1.0-SSQ*TM14
      IF(NFLAG .EQ. 0) GO TO 70
      T11 = -S*TM11
      T44 = -S*TM44
      IF(ISO .NE. 0) RETURN
      T31 = TM31
      T12 = S*TM12
      T42 = TM42
      T34 = S*TM34
      RETURN
C
C
      ENTRY TI MTRX
      T41 = TM41
      T32 = CSQI+TM32
      T14 = 1.0-SSQI*TM14
      T11 = -SI*TM11
      T44 = -SI*TM44
      IF(ISO .NE. 0) RETURN
      T31 = TM31
      T12 = SI*TM12
      T42 = TM42
      T34 = SI*TM34
      RETURN
C
C
      ENTRY INIT T
C   COMPUTE VARIOUS QUANTITIES
C  WHICH DO NOT VARY WITH HEIGHT.
      LHT = 0
      ISO=0
      IF(MAGFLD .EQ. 0.0) GO TO 250
      IF (DABS(CODIP-90.0).GE.0.15)  GO TO 300
      IF (DABS(AZMUTH-90.0).LT.0.15)  GO TO 250
      IF (DABS(AZMUTH-270.0).GE.0.15)  GO TO 300
  250 ISO = 1
300   OMEGA = 2000.0*PI*FREQ
      COMGA = 1.0/OMEGA
      WAVENR = OMEGA/VELLT
      SINDIP = DSIN (CODIP*DTR)
      DRCOSL = SINDIP * DCOS (AZMUTH * DTR)
      DRCOSM =-SINDIP * DSIN (AZMUTH * DTR)
      DRCOSN =    DCOS (CODIP * DTR)
      DO 60 K=1,NRSPEC
      COEFEN(K) = COEFFX*1.0E6*CHARGE(K)**2/(OMEGA**2*RATIOM(K))
      Y(K) = COEFFY * CHARGE(K) * MAGFLD
     $ / (OMEGA * RATIOM(K))
      YSQ(K)=Y(K)**2
      LY(K) = DRCOSL*Y(K)
```

```
      MY(K) = DRCOSM*Y(K)
      NY(K) = DRCOSN*Y(K)
      LSQYSQ(K)=DRCOSL**2*YSQ(K)
      MSQYSQ(K)=DRCOSM**2*YSQ(K)
      NSQYSQ(K)=DRCOSN**2*YSQ(K)
      LMYSQ(K)=DRCOSL*DRCOSM*YSQ(K)
      LNYSQ(K)=DRCOSL*DRCOSN*YSQ(K)
      MNYSQ(K)=DRCOSM*DRCOSN*YSQ(K)
   60 CONTINUE
      C = CDCOS(THETA*DTR)
      S = CDSIN(THETA*DTR)
      CSQ = C**2
      SSQ = S**2
      CI = CDCOS((THETA-DTHETA)*DTR)
      SI = CDSIN((THETA-DTHETA)*DTR)
      CSQI = CI**2
      SSQI = SI**2
C
   70 T11 = 0.0
      T31 = 0.0
      T12 = 0.0
      T42 = 0.0
      T34 = 0.0
      T44 = 0.0
      RETURN
      END
```

```
      SUBROUTINE WF DENS (HT, EN, COLL)
C
C   WF DENS COMPUTES THE ION DENSITY
C   AND COLLISION FREQUENCY FOR EACH
C   SPECIE BY LOGARITHMIC INTERPOLATION
C   OF THE CORRESPONDING PROFILES.
C   PROFILE VALUES ARE INTERPOLATED BETWEEN
C   ENTRIES MHT AND MHT+1 (LHT AND LHT+1).
C
      IMPLICIT REAL *8 (A-H,O-Z)
      COMMON/WFPROF/ENHT(100),ENLOG(100,5),COLLHT(25),COLLFR(25,5),
     $              LHT,MHT,CHARGE(5),RATIOM(5),NRSPEC
      DIMENSION  EN(5), COLL(5), DELE(5), DELC(5)
      COMMON/PRIMES/DELE,DELC
      DATA EPSHT/5.D-4/
C
      LUCKY=0
      MUCKY=0
      IF (LHT.EQ.0) LHT=1
      IF (MHT.EQ.0) MHT=1

   10 IF (HT.GE.ENHT(LHT+1)-EPSHT .AND. HT.LT.ENHT(LHT)+EPSHT) GO TO 20
      IF (LUCKY.EQ.1) GO TO 30
      LHT=LHT-1
      IF (LHT.EQ.0) LHT=1
      IF (LHT.EQ.1) LUCKY=1
      GO TO 10
   30 LHT=LHT+1
      IF (LHT.GT.101) GO TO 899
      GO TO 10

   20 IF(HT.GE.COLLHT(MHT+1)-EPSHT.AND.HT.LT.COLLHT(MHT)+EPSHT)GOTO 100
      IF (MUCKY.EQ.1) GO TO 40
      MHT=MHT-1
      IF (MHT.EQ.0) MHT=1
      IF (MHT.FQ.1) MUCKY=1
      GO TO 20
   40 MHT=MHT+1
      IF (MHT.GT. 26) GO TO 899
      GO TO 20
C
  100 IF (LHT.EQ.LSAVE) GO TO 200
      DO 150  K = 1,NRSPEC
      DELE(K) = (ENLOG(LHT+1,K) - ENLOG(LHT,K))
     $ / (ENHT(LHT+1) - ENHT(LHT))
  150 CONTINUE
      LSAVE = LHT
C
  200 IF (MHT.EQ.MSAVE)  GO TO 300
      DO 250  K = 1,NRSPEC
      DELC(K) = (COLLFR(MHT+1,K) - COLLFR(MHT,K))
     $ / (COLLHT(MHT+1) - COLLHT(MHT))
  250 CONTINUE
      MSAVE = MHT
```

```
300 DH = HT - ENHT(LHT)
    DC = HT - COLLHT(MHT)
    DO 500  K = 1,NRSPEC
    EN(K) = DEXP (ENLOG(LHT,K) + DH * DELE(K))
    COLL(K)  = DEXP (COLLFR(MHT,K) + DC * DELC(K))
500 CONTINUE
    RETURN
899 PRINT 900
900 FORMAT (' ERROR IN PROFILE INTERPOLATION')
    STOP
    END
```

```
      SUBROUTINE WF INIT(P)
C
C  WF INIT COMPUTES THE INITIAL
C  P MATRIX, I.E., THE INITIAL CONDITIONS
C  FOR THE INTEGRATION DP/DZ = -IK*T*P.
C
      COMMON/CS/C,IGNORE(12)
      COMPLEX*16 P(4,2),C
C
      P(1,1) = C
      P(2,1) = 0.
      P(3,1) = 0.
      P(4,1) = 1.
      P(1,2) = 0.
      P(2,2) = 1.
      P(3,2) = C
      P(4,2) = 0.
C
      END
```

```
      SUBROUTINE P DERIV(P,DPDH)
C
C P DERIV COMPUTES THE HEIGHT DERIVATIVES
C OF THE FIELD VECTORS P ACCORDING TO
C CLEMMOW AND HEADING (1954).
C EQUATION IS DP/DZ = -IK*T*P.
C MULTIPLICATION BY -I IS PERFORMED BY
C OPERATING ON REAL AND IMAG PARTS.
C MULTIPLICATION BY K IS PERFORMED
C IN SUBROUTINE WF STEP.
C
      COMMON/T MTX/ T11,T31,T41,T12,T32,T42,T14,T34,T44
      COMPLEX*16 P(4,2),DPDH(4,2),DERIV,
     $           T11,T31,T41,T12,T32,T42,T14,T34,T44
C
C
      DO 11 J=1,2
      DERIV = T11*P(1,J)+T12*P(2,J)+T14*P(4,J)
      DPDH(1,J) = (0.D0,-1.D0)*DERIV
      DERIV = P(3,J)
      DPDH(2,J) = (0.D0,-1.D0)*DERIV
      DERIV = T31*P(1,J)+T32*P(2,J)+T34*P(4,J)
      DPDH(3,J) = (0.D0,-1.D0)*DERIV
      DERIV = T41*P(1,J)+T42*P(2,J)+T44*P(4,J)
      DPDH(4,J) = (0.D0,-1.D0)*DERIV
   11 CONTINUE
      RETURN
C
      END
```

```
      SUBROUTINE XFER(A,B,N)
C
C  TRANSFER ARRAY A INTO ARRAY B.
C
      COMPLEX*16 A,B
      DIMENSION A(1),B(1)
C
C
      DO 11 J=1,N
   11 B(J) = A(J)
      RETURN
C
      END
```

```
      SUBROUTINE WF SCAL(PP,IFLAG)
C
C   WFSCAL SCALES AND ORTHOGONALIZES THE SOLUTION
C   VECTORS P.  THIS SCALING MUST LATER BE
C   REMOVED TO YIELD CORRECT (UNSCALED) SOLUTIONS.
C
      IMPLICIT  REAL*8 (A-H,O-Z)
      COMMON/WF SAVE/P SAVE(8),M31 SAV,M32 SAV,M33 SAV,
     $          O SUM,APROD,BPROD,HT,LEVL
      COMMON/SAVE/P ETC(27,129)
      COMPLEX*16 P(4,2),PP,PSAVE,
     $                 M31 SAV,M32 SAV,M33 SAV,O SUM,ORTHO
      DIMENSION PR(8,2),PP(8),PSTORE(27)
      EQUIVALENCE (PSAVE,PSTORE)
      EQUIVALENCE (P,PR)
C
C
      CALL XFER(PP,P,8)
      ANORM = 0.0
      DO 11 J=1,8
   11 ANORM = ANORM+PR(J,1)**2
      ORTHO = 0.0
      DO 12 J=1,4
   12 ORTHO = ORTHO+DCONJG(P(J,1))*P(J,2)
      ORTHO = ORTHO/ANORM
C
      DO 13 J=1,4
   13 P(J,2) = P(J,2)-ORTHO*P(J,1)
      BNORM = 0.0
      DO 14 J=1,8
   14 BNORM = BNORM+PR(J,2)**2
C
      ANORM = 1.0/DSQRT(ANORM)
      BNORM = 1.0/DSQRT(BNORM)
      DO 15 J=1,8
      PR(J,1) = PR(J,1)*ANORM
   15 PR(J,2) = PR(J,2)*BNORM
      CALL XFER(P,PP,8)
      IF(IFLAG .NE. 0) RETURN
C
      O SUM = O SUM+ORTHO*APROD/BPROD
      APROD = APROD*ANORM
      BPROD = BPROD*BNORM
      RETURN
C
C
      ENTRY WF STOR
      LEVL = LEVL+1
      DO 25 J=1,27
   25 P ETC(J,LEVL)=PSTORE(J)
      O SUM = 0.0
      A PROD = 1.0
      B PROD = 1.0
      RETURN
```

```
C
C
      ENTRY WF LOAD
      DO 35 J=1,27
   35 PSTORE(J)=P ETC(J,LEVL)
      LEVL = LEVL-1
      RETURN
C
      END
```

```
      SUBROUTINE WF STEP(P,DPDH,HT,DELH,IFLAG)
C
C  WF STEP INCREMENTS THE SOLUTION OF P
C  FROM HT TO HT-DELH, USING
C  RUNGE-KUTTA INTEGRATION
C
C
C  IFLAG=0  ONE LARGE STEP, THETA
C  IFLAG=1  FIRST SMALL STEP, THETA
C  IFLAG=2  SECOND SMALL STEP, THETA
C  IFLAG=3  FIRST SMALL STEP, THETA-DTHETA
C  IFLAG=4  SECOND SMALL STEP, THETA-DTHETA
C
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/WF CON/OMEGA,WAVE NR
      COMMON/WF FLAG/PRECSN,ISO,IDBG
      COMMON/T MTX/T(9)
      COMMON/TM MTX/TM(9)
      COMPLEX*16 P(8),DPDH(8),PO(8),HDELPO(8),DELP1(8),DELP2(8),DELP4
      COMPLEX*16  T,TSAVE1(9),TSAVE2(9),
     $            TM,TMSAV1(9),TMSAV2(9),TMSAV3(9),TMSAV4(9)
C
C
      HTO = HT
      DELH K = DELH*WAVE NR
      HDELH K = DELH K*0.5
      DO 11 J=1,8
      PO(J) = P(J)
      HDELPO(J) = -DPDH(J)*HDELH K
   11 P(J) = PO(J)+HDELPO(J)
C
      HT = HTO-0.5*DELH
      IF(IFLAG .LE. 2) CALL T MTRX(HT)
      IF(IFLAG .EQ. 0) CALL XFER(T,T SAVE1,9)
      IF(IFLAG .EQ. 0) CALL XFER(TM,TM SAV2,9)
      IF(IFLAG .EQ. 1) CALL XFER(TM,TM SAV1,9)
      IF(IFLAG .EQ. 2) CALL XFER(TM,TM SAV3,9)
      IF(IFLAG .EQ. 3) CALL XFER(TM SAV1,TM,9)
      IF(IFLAG .EQ. 4) CALL XFER(TM SAV3,TM,9)
      IF(IFLAG .GE. 3) CALL TI MTRX
C
      CALL P DERIV(P,DPDH)
      DO 12 J=1,8
      DELP1(J) = -DPDH(J)*DELH K
   12 P(J) = PO(J)+0.5*DELP1(J)
C
      CALL P DERIV(P,DPDH)
      DO 13 J=1,8
      DELP2(J) = -DPDH(J)*DELH K
   13 P(J) = PO(J)+DELP2(J)
C
      HT = HTO-DELH
```

```
      IF(IFLAG .EQ. 0) CALL T MTRX(HT)
      IF(IFLAG .EQ. 0) CALL XFER(T,T SAVE2,9)
      IF(IFLAG .EQ. 1) CALL XFER(T SAVE1,T,9)
      IF(IFLAG .EQ. 2) CALL XFER(T SAVE2,T,9)
      IF(IFLAG .EQ. 0) CALL XFER(TM,TM SAV4,9)
      IF(IFLAG .EQ. 3) CALL XFER(TM SAV2,TM,9)
      IF(IFLAG .EQ. 4) CALL XFER(TM SAV4,TM,9)
      IF(IFLAG .GE. 3) CALL TI MTRX
C
      CALL P DERIV(P,DPDH)
      THIRD = 1.0D0/3.0D0
      DO 14 J=1,8
      DELP4 = (HDELP0(J)+DELP1(J)+DELP2(J)-DPDH(J)*HDELH K)*THIRD
   14 P(J) = P0(J)+DELP4
      RETURN
C
      END
```

```
      SUBROUTINE WKB
      IMPLICIT REAL*8(A-H,O-Z)
      INTEGER STPFLG
      REAL*8 MAGFLD,LWSTHT
      COMPLEX*16 I,THETA,Q,GAMMA,INTGRQ,INTGAM,SUMQ,SUMGAM,EXPO,
     $   P(4,2),PARTS
      COMMON/WFINPT/THETA,FREQ,AZIM,CODIP,MAGFLD,COEFNU(5),EXPNU(5),
     $                 TOPHT,LWSTHT,WKBHT,DELHT,H,ALPHA,SIGMA,EPSLON
      COMMON/WFCON/OMEGA,WAVENR
      COMMON/DELTA/NUMDIV,STPFLG
      COMMON/ANSWER/PART S(4,2)
      COMMON/P MTX/P,IGNORR(32)
      DATA I/(0.0D0,1.0D0)/

      IF (WKBHT.GE.LWSTHT) RETURN
      CALL INIT DT
C  STPFLG=0    FIXED INTEGRATION STEPSIZE OF DELHT/NUMDIV
C  STPFLG=1    VARIABLE INTEGRATION STEPSIZE
      IF (STPFLG.EQ.1) CALL WKBVAR
      IF (STPFLG.EQ.1) RETURN

      EPSHT=DELHT/NUMDIV
      IHT=(LWSTHT-WKBHT)/EPSHT + 1.01D0
      SUMQ=(0.0D0,0.0D0)
      SUMGAM=(0.0D0,0.0D0)

      DO 90 N=1,IHT-2,2
      DO 30 M=1,3
      HT = LWSTHT - EPSHT*(N+M-2)
      CALL QGAMMA(HT,DELHT,LWSTHT,Q,GAMMA)
      IF (M.EQ.2) GO TO110
      INTGRQ=EPSHT*Q/3.0D0
      INTGAM=EPSHT*GAMMA/3.0D0
      GO TO 120
  110 INTGRQ=4.0D0*EPSHT*Q/3.0D0
      INTGAM=4.0D0*EPSHT*GAMMA/3.0D0
  120 SUMQ=SUMQ+INTGRQ
   30 SUMGAM=SUMGAM+INTGAM

      NNN=(N+1)/NUMDIV
      IF ((N+1)/NUMDIV - NNN .NE. 0.0) GO TO 90
      EXPO=CDEXP(WAVENR*(SUMQ*I - SUMGAM))
      DO  150 K=1,4
      DO 150 L=1,2
  150 P(K,L) = EXPO*PART S(K,L)
C     OUTPUT OF INTERMEDIATE P VALUES COULD BE ACCOMPLISHED HERE
   90 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE WFBVAR
      IMPLICIT REAL*8(A-H,O-Z)
      INTEGER STPFLG,SVFLAG
      REAL*8 MAGFLD,LWSTHT
      COMPLEX*16 I,THETA,Q,GAMMA,INTGRQ,INTGAM,SUMQ,SUMGAM,EXPO,QSAV,
     $     (4,2),PARTS
      COMMON WFINPT THETA,FREQ,AZIM,CODIP,MAGFLD,COEFNU(5),EXPNU(5),
     $                 TOPHT,LWSTHT,WKBHT,DELHT,H,ALPHA,SIGMA,EPSLON
      COMMON WFCON OMEGA,WAVENR
      COMMON DELTA NUMDIV,STPFLG
      COMMON WFFLAG PRECSN,ISO,IDBUG
      COMMON ANSWER,PART S(4,2)
      COMMON RMTX P,IGNORR(32)
      DATA EPSHT 5.D-4/
      DATA DHMIN 1.D-3/
      DATA I/(0.0D0,1.0D0)/

      SVFLAG = 0
      SUMQ=(0.0D0,0.0D0)
      SUMGAM=(0.0D0,0.0D0)
      NUMINT=0
      HT=LWSTHT
      WFHT=LWSTHT-DELHT
      DELH2=DELHT/8.
      DELHSV - DELH2

   10 IF (SVFLAG.EQ.1) DELH2=SAVDH2
      SVFLAG=0
      HTLIM=WFHT
      NODBL=0
      IF (HT-DELH2.GE.HTLIM-EPSHT) GO TO 50
      SAVDH2=DELH2
      SVFLAG=1
      DELH2=HT-HTLIM

   50 LFLAG=0
   60 INTGRQ=(0.,0.)
      INTGAM=(0.,0.)
      IF(DELH2  .LT. DELHSV) DELHSV=DELH2
   65 DO 30 M=1,3
      CALL QGAMMA(HT,DELHT,LWSTHT,Q,GAMMA)
      IF (M.EQ.2) GO TO110
      INTGRQ=DELH2*Q/6. + INTGRQ
      INTGAM=DELH2*GAMMA/6. + INTGAM
      GO TO 30
  110 INTGRQ=2.*DELH2*Q/3. + INTGRQ
      INTGAM=2.*DELH2*GAMMA/3. + INTGAM
   30 IF (M.NE.3) HT=HT-DELH2/2.

      LFLAG=LFLAG+1
      IF(LFLAG.EQ.3) GO TO 70
      IF (LFLAG.EQ.2) GO TO 65
      QSAV=INTGRQ
      HT=HT+DELH2
```

```
      DELH2=DELH2/2.
      GO TO 60
   70 DELH2=DELH2*2.
      QABS=CDABS(QSAV-INTGRQ)
      IF (CDABS(INTGRQ).NE.0.) QABS=QABS/CDABS(INTGRQ)
      IF (QABS.LT.PRECSN) GO TO 100
      IF (DELH2.GT.DHMIN) GO TO 95
      IF (KMAX.EQ.0) PRINT 900,HT
  900 FORMAT (' MINIMUM STEPSIZE USED FIRST AT HT =',D14.5)
      KMAX=1
      GO TO 100
   95 HT=HT+DELH2
      DELH2=DELH2/2.
      NODBL=1
      IF (QABS.LT.10.*PRECSN) GO TO 99
      DELH2=DELH2/4.
      NODBL=0
   99 SVFLAG=0
      GO TO 50

  100 IF (NODBL.EQ.0 .AND. QABS.LT.PRECSN/3.) DELH2=2.*DELH2
  120 SUMQ=SUMQ+INTGRQ
      SUMGAM=SUMGAM+INTGAM
      NUMINT=NUMINT+1
      IF (HT.GT.WFHT+EPSHT) GO TO 10
      WFHT=WFHT-DELHT
      EXPO=CDEXP(WAVENR*(SUMQ*I - SUMGAM))
      DO  150 K=1,4
      DO 150 L=1,2
  150 P(K,L) = EXPO*PART S(K,L)
C     OUTPUT OF INTERMEDIATE P VALUES COULD BE ACCOMPLISHED HERE
      IF (HT.GT.WKBHT+EPSHT) GO TO 10
      DELHAV = (LWSTHT-WKBHT)/(4*NUMINT)
      PRINT 909,LWSTHT,WKBHT,DELHSV
      PRINT 910,LWSTHT,WKBHT,DELHAV
      RETURN
909   FORMAT(' SMALLEST INTEGRATION INTERVAL BETWEEN LWSTHT'' = ',F7.1,
     $  ' KM AND WKBHT'' = ',F7.1,' KM IS ',F9.4,' KM')
910   FORMAT(' AVERAGE  INTEGRATION INTERVAL BETWEEN LWSTHT'' = ',F7.1,
     $  ' KM AND WKBHT'' = ',F7.1,' KM IS ',F9.4,' KM')
      END
```

```
      SUBROUTINE [illegible]AMTEHT,M,T,MPRIME,TPRIME)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON PRIVES DELE,DELC
      COMMON WF FLAG FRESON,ISJ,IDBG
      COMMON CS C,S,CI,SI
      COMMON WFINPT THETA,FREQ,AZMUTH,CODIF,MAGFLD,CEFFNU(5),EXFNU(5),
     $ TOPHT,LWSTHT,WRBHT,DELHT,H,ALPHA,SIGMA,EPSLON
      COMMON WF DOJ OMEGA,WAVE NR
      COMMON WFPROF ENHT(100),ENLOG(100,5),COLLHT(25),COLLFR(25,5),
     $                  LHT,HHT,CHARGE(5),RATIOM(5),NRSPEC
      REAL*8  MAGFLD, LWSTHT,
     $          LSQYSQ, MSQYSQ, NSQYSQ,
     $ LMYSQ, LNYSQ, MNYSQ, NU,
     $            LY,MY,NY
      COMPLEX*16 M(3,3),
     $     T(4,4),
     $     MPRIME(3,3),TPRIME(4,4),XPRIME,UPRIME,DDPRIM,IUDPRI,M331,
     $     TAPRIM,TBPRIM,
     $            C,S,CI,SI,CSQ,SSQ,CSQI,SSQI,
     $            THETA,DTHETA,
     $            U,USQ,DD,I,IUD,TA,TB
      DIMENSION DELE(5),DELC(5),ENPRIM(5),NUPRIM(5)
      DIMENSION  Y(5), YSQ(5),  LY(5),  MY(5),
     $  NY(5), LMYSQ(5), LNYSQ(5), MNYSQ(5), EN(5), NU(5),
     $ LSQYSQ(5), MSQYSQ(5), NSQYSQ(5),
     $            COEF EN(5)
C
      DATA PI/3.141592653D0/
      DATA DTR/1.745329252D-2/
      DATA COEFFX/3.182357D03/,COEFFY/1.758796D11/
      DATA I/(0.0D0,1.0D0)/
      DATA VELLT/2.997928D05/
      DATA DTHETA/(5.0D-2,1.0D-2)/
C
C
C    CALCULATE THE MATRIX M.
      M(1,1) = 0.0
      M(1,2) = 0.0
      M(1,3) = 0.0
      M(2,1) = 0.0
      M(2,2) = 0.0
      M(2,3) = 0.0
      M(3,1) = 0.0
      M(3,2) = 0.0
      M(3,3) = 0.0
      MPRIME(1,1)=0.0
      MPRIME(1,2)=0.0
      MPRIME(1,3)=0.0
      MPRIME(2,1)=0.0
      MPRIME(2,2)=0.0
      MPRIME(2,3)=0.0
      MPRIME(3,1)=0.0
      MPRIME(3,2)=0.0
      MPRIME(3,3)=0.0
```

```
C
      CALL WF DENS (HT, EN, NU)
      DO 10 K=1,NRSPEC
      ENPRIM(K)=EN(K)*DELE(K)
      NUPRIM(K)=NU(K)*DELC(K)
   10 CONTINUE
C
      NFLAG = 0
      DO 20 K=1,NRSPEC
C   ADD IN THE CONTRIBUTIONS TO THE
C   SUSCEPTIBILITY TENSOR M FOR EACH
C   SPECIE IN THE IONOSPHERE.
      IF(EN(K) .LT. 1.0E-3) GO TO 20
      NFLAG = 1
      X = COEF EN(K)*EN(K)
      XPRIME=COEFEN(K)*ENPRIM(K)/WAVENR
      Z = NU(K)*OV OMGA
      U=1.0-I*Z
      UPRIME=-I*NUPRIM(K)*OV OMGA/WAVENR
      USQ=U*U
      DD = -X / (U * (USQ - YSQ(K)) )
      DDPRIM=(X*(3.D0*USQ*UPRIME-YSQ(K)*UPRIME) -
     1        XPRIME*(U*(USQ-YSQ(K))) )/(U*(USQ-YSQ(K)))**2
      IUD = (Z+I)*DD
      IUDPRI=NUPRIM(K)*OV OMGA/WAVENR*DD     + (Z+I)*DDPRIM

      TA = USQ * DD
      M(1,1) = M(1,1) + TA
      M(2,2) = M(2,2) + TA
      M(3,3) = M(3,3) + TA
      M(2,2) = M(2,2) - MSQYSQ(K) * DD
      TAPRIM=2.D0*U*UPRIME*DD + USQ*DDPRIM
      MPRIME(1,1)=MPRIME(1,1) + TAPRIM
      MPRIME(2,2)=MPRIME(2,2) + TAPRIM
      MPRIME(3,3)=MPRIME(3,3) + TAPRIM
      MPRIME(2,2)=MPRIME(2,2) - MSQYSQ(K) * DDPRIM

      TA = MY(K)*IUD
      TB = LNYSQ(K) * DD
      M(1,3) = M(1,3) + TA - TB
      M(3,1) = M(3,1) - TA - TB
      TAPRIM=MY(K)*IUDPRI
      TBPRIM=LNYSQ(K)*DDPRIM
      MPRIME(1,3)=MPRIME(1,3) + TAPRIM - TBPRIM
      MPRIME(3,1)=MPRIME(3,1) - TAPRIM - TBPRIM

      IF (ISO.NE.C)  GO TO 20
      M(1,1) = M(1,1) - LSQYSQ(K) * DD
      M(3,3) = M(3,3) - NSQYSQ(K) * DD
      MPRIME(1,1)=MPRIME(1,1) - LSQYSQ(K)*DDPRIM
      MPRIME(3,3)=MPRIME(3,3) - NSQYSQ(K)*DDPRIM
      TA = NY(K)*IUD
      TB = LMYSQ(K) * DD
      M(2,1) = M(2,1) + TA - TB
      M(1,2) = M(1,2) - TA - TB
      TAPRIM=NY(K)*IUDPRI
      TBPRIM=LMYSQ(K)*DDPRIM
```

```
      MPRIME(2,1)=MPRIME(2,1) + TAPRIM - TBPRIM
      MPRIME(1,2)=MPRIME(1,2) - TAPRIM - TBPRIM
      TA = LY(K)*IUD
      TB = WNYSQ(K) * DD
      M(3,2) = M(3,2) + TA - TB
      M(2,3) = M(2,3) - TA - TB
      TAPRIM=LY(K)*IUDPRI
      TBPRIM=WNYSQ(K)*DDPRIM
      MPRIME(3,2)=MPRIME(3,2) + TAPRIM -TBPRIM
      MPRIME(2,3)=MPRIME(2,3) - TAPRIM -TBPRIM
   20 CONTINUE
C
      CRVTRM=ALPHA*(H-HT)
      M(1,1) = M(1,1) + CRVTRM
      M(2,2) = M(2,2) + CRVTRM
      M(3,3) = M(3,3) + CRVTRM
      CURVPR=-ALPHA/WAVENR
      MPRIME(1,1)=MPRIME(1,1) + CURVPR
      MPRIME(2,2)=MPRIME(2,2) + CURVPR
      MPRIME(3,3)=MPRIME(3,3) + CURVPR
C
C   CALCULATE THE MATRIX T.

      T(1,1)=-S*M(3,1)/(1.0D0+M(3,3))
      T(4,4)=-S*M(1,3)/(1.0D0+M(3,3))
      T(1,2)= S*M(3,2)/(1.0D0+M(3,3))
      T(3,4)= S*M(2,3)/(1.0D0+M(3,3))
      T(1,3)=0.0D0
      T(2,4)=0.0D0
      T(1,4)=(C*C+M(3,3))/(1.0D0+M(3,3))
      T(2,1)=0.0D0
      T(4,3)=0.0D0
      T(2,2)=0.0D0
      T(3,3)=0.0D0
      T(2,3)=1.0D0
      T(3,1)=M(2,3)*M(3,1)/(1.0D0+M(3,3)) - M(2,1)
      T(4,2)=M(3,2)*M(1,3)/(1.0D0+M(3,3)) - M(1,2)
      T(3,2)= C*C + M(2,2) - M(2,3)*M(3,2)/(1.0D0+M(3,3))
      T(4,1)=1.D0 + M(1,1) - M(1,3)*M(3,1)/(1.0D0+M(3,3))

      M331=1.0D0+M(3,3)
      TPRIME(1,1)=(-S*MPRIME(3,1)*M331+S*M(3,1)*MPRIME(3,3))/M331**2
      TPRIME(4,4)=(-S*MPRIME(1,3)*M331+S*M(1,3)*MPRIME(3,3))/M331**2
      TPRIME(1,2)=( S*MPRIME(3,2)*M331-S*M(3,2)*MPRIME(3,3))/M331**2
      TPRIME(3,4)=( S*MPRIME(2,3)*M331-S*M(2,3)*MPRIME(3,3))/M331**2
      TPRIME(1,3)=0.0D0
      TPRIME(2,4)=0.0D0
      TPRIME(1,4)=( MPRIME(3,3)*M331 - MPRIME(3,3)*(C*C+M(3,3)))/M331**2
      TPRIME(2,1)=0.0D0
      TPRIME(4,3)=0.0D0
      TPRIME(2,2)=0.0D0
      TPRIME(3,3)=0.0D0
      TPRIME(2,3)=0.0D0
      TPRIME(3,1)=( M(2,3)*MPRIME(3,1)*M331+MPRIME(2,3)*M(3,1)*M331 -
     1              MPRIME(3,3)*M(2,3)*M(3,1) )/M331**2 - MPRIME(2,1)
      TPRIME(4,2)=( M(3,2)*MPRIME(1,3)*M331+MPRIME(3,2)*M(1,3)*M331 -
     1              MPRIME(3,3)*M(3,2)*M(1,3) )/M331**2 - MPRIME(1,2)
```

```
      TPRIME(3,2)=MPRIME(2,2) - (MPRIME(2,3)*M(3,2)*M331+
     1            M(2,3)*MPRIME(3,2)*M331-MPRIME(3,3)*M(2,3)*M(3,2) )/
     2            M331**2
      TPRIME(4,1)=MPRIME(1,1) - (MPRIME(1,3)*M(3,1)*M331+
     1            M(1,3)*MPRIME(3,1)*M331-MPRIME(3,3)*M(1,3)*M(3,1) )/
     2            M331**2
      RETURN
C
C
      ENTRY INIT DT
      ISO=0
      IF(MAGFLD .EQ. 0.0) GO TO 250
      IF (DABS(CODIP-90.0).GE.0.15)  GO TO 300
      IF (DABS(AZMUTH-90.0).LT.0.15)  GO TO 250
      IF (DABS(AZMUTH-270.0).GE.0.15)  GO TO 300
  250 ISO = 1
  300 OMEGA = 2000.0*PI*FREQ
      OV OMGA = 1.0/OMEGA
      WAVENR = OMEGA/VELLT
      SINDIP = DSIN (CODIP*DTR)
      DRCOSL = SINDIP * DCOS (AZMUTH * DTR)
      DRCOSM =-SINDIP * DSIN (AZMUTH * DTR)
      DRCOSN =    DCOS (CODIP * DTR)
      DO 60 K=1,NRSPEC
      COEF EN(K) = COEFFX*1.0E6*CHARGE(K)**2/(OMEGA**2*RATIOM(K))
      Y(K) = COEFFY * CHARGE(K) * MAGFLD
     $ / (OMEGA * RATIOM(K))
      YSQ(K)=Y(K)**2
      LY(K) = DRCOSL*Y(K)
      MY(K) = DRCOSM*Y(K)
      NY(K) = DRCOSN*Y(K)
      LSQYSQ(K)=DRCOSL**2*YSQ(K)
      MSQYSQ(K)=DRCOSM**2*YSQ(K)
      NSQYSQ(K)=DRCOSN**2*YSQ(K)
      LMYSQ(K)=DRCOSL*DRCOSM*YSQ(K)
      LNYSQ(K)=DRCOSL*DRCOSN*YSQ(K)
      MNYSQ(K)=DRCOSM*DRCOSN*YSQ(K)
   60 CONTINUE
      C = CDCOS(THETA*DTR)
      S = CDSIN(THETA*DTR)
      CSQ = C**2
      SSQ = S**2
      CI = CDCOS((THETA-DTHETA)*DTR)
      SI = CDSIN((THETA-DTHETA)*DTR)
      CSQI = CI**2
      SSQI = SI**2
C
      RETURN
      END
```

```
      SUBROUTINE QGAMMA (HT,DELHT,LWSTHT,Q,GAMMA)
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 LWSTHT
      COMPLEX*16 C,S,CI,SI,I,QQ(4),Q,GAMMA,AJ,FJ,F,DERIVF,
     $   M(3,3),T(4,4),MPRIME(3,3),TPRIME(4,4),
     $   COEFFB,COEFFC,COEFFD,COEFFE,COEFF0,COEFF1,COEFF2,
     $   A1,A2,A3,A4,A5,A6,B3,B4,B5,B6,
     $   A3PRIM,A4PRIM,A5PRIM,A6PRIM,
     $   B3PRIM,B4PRIM,B5PRIM,B6PRIM,
     $   SPVEC,S NORM(4,2), PART S
      COMMON/CS/C,S,CI,SI
      COMMON/ANSWER/PART S(4,2)
      COMMON/S MTX/SPVEC(4,4)
      DATA I/(0.0D0,1.0D0)/

      CALL DDKXMT(HT,M,T,MPRIME,TPRIME)
      COEFFA = 1.0D0
      COEFFB = -(T(1,1)+T(4,4))
      COEFFC = T(1,1)*T(4,4) - T(1,4)*T(4,1) - T(3,2)
      COEFFD = T(3,2)*(T(1,1)+T(4,4)) - T(1,2)*T(3,1) - T(3,4)*T(4,2)
      COEFFE = T(1,1)*(T(3,4)*T(4,2) - T(3,2)*T(4,4))  +
     1         T(1,2)*(T(3,1)*T(4,4) - T(3,4)*T(4,1))  +
     2         T(1,4)*(T(3,2)*T(4,1) - T(3,1)*T(4,2))

      JJ=7
      IF (HT.LT.LWSTHT) GO TO 400
   10 CALL QUARTC (COEFFB,COEFFC,COEFFD,COEFFE,QQ)
      QR MAX=QQ(1)
      J2=1
      DO 23 J=2,4
      QREAL=QQ(J)
      IF (QREAL.LT.QR MAX) GO TO 23
      QRMAX=QREAL
      J2=J
   23 CONTINUE
      Q=QQ(J2)

  400 K=1
  500 CONTINUE
      IF (K.GT.21 ) GO TO 510
      F = Q**4*COEFFA + Q**3*COEFFB + Q**2*COEFFC + Q*COEFFD + COEFFE
      DERIVF=4.D0*Q**3*COEFFA + 3.D0*Q**2*COEFFB + 2.D0*Q*COEFFC +COEFFD
      REAL=F/DERIVF
      GINARY=-I*F/DERIVF
      IF  ((REAL**2.LE.1.D-20).AND.(GINARY**2.LE.1.D-30)) GO TO 530
      Q=Q-F/DERIVF
      K=K+1
      GO TO 500
  510 IF (JJ.EQ.7) GO TO 570
      PRINT 520,HT
  520 FORMAT (10X,'DOES NOT CONVERGE AFTER 20  ITERATIONS AT HEIGHT=',
     1  F10.5/)
  530 CONTINUE
      REAL=Q
```

```
      GINARY=-I*Q
      IF (REAL**2 .GT. GINARY**2 .AND. REAL.GT.0.) GO TO 540
      IF (JJ.EQ.13) GO TO 550
  570 JJ=13
      GO TO 10
  550 PRINT 560, HT
  560 FORMAT (10X,'DOES NOT CONVERGE TO THE PROPER Q AT HEIGHT=',F10.5/)
  540 CONTINUE

      A1=-(T(1,1)+T(4,4))
      A2=T(1,1)*T(4,4) - T(1,4)*T(4,1)
      A3=T(1,2)
      A4=T(1,4)*T(4,2) - T(1,2)*T(4,4)
      A5=T(4,2)
      A6=T(4,1)*T(1,2) - T(1,1)*T(4,2)
      B3=T(3,4)
      B4=T(1,4)*T(3,1) - T(3,4)*T(1,1)
      B5=T(3,1)
      B6=T(4,1)*T(3,4) - T(4,4)*T(3,1)
      A3PRIM=TPRIME(1,2)
      A4PRIM=TPRIME(1,4)*T(4,2)+T(1,4)*TPRIME(4,2) -
     1        TPRIME(1,2)*T(4,4)-T(1,2)*TPRIME(4,4)
      A5PRIM=TPRIME(4,2)
      A6PRIM=TPRIME(4,1)*T(1,2)+T(4,1)*TPRIME(1,2) -
     1        TPRIME(1,1)*T(4,2)-T(1,1)*TPRIME(4,2)
      B3PRIM=TPRIME(3,4)
      B4PRIM=TPRIME(1,4)*T(3,1)+T(1,4)*TPRIME(3,1) -
     1        TPRIME(3,4)*T(1,1)-T(3,4)*TPRIME(1,1)
      B5PRIM=TPRIME(3,1)
      B6PRIM=TPRIME(4,1)*T(3,4)+T(4,1)*TPRIME(3,4) -
     1        TPRIME(4,4)*T(3,1)-T(4,4)*TPRIME(3,1)
      AJ=Q*Q+A1*Q+A2
      FJ=2.0D0*Q*AJ + (Q*Q-T(3,2))*(2.0D0*Q+A1) - (A3*B5+B3*A5)
      COEFF0=A4*B6PRIM-A4PRIM*B6+A6*B4PRIM-A6PRIM*B4
      COEFF1=A3*B6PRIM-A3PRIM*B6+A4*B5PRIM-A4PRIM*B5+A5*B4PRIM-
     1        A5PRIM*B4+A6*B3PRIM-A6PRIM*B3
      COEFF2=A3*B5PRIM-A3PRIM*B5+A5*B3PRIM-A5PRIM*B3
      GAMMA=(Q*Q*COEFF2 + Q*COEFF1 + COEFF0)/(2.D0*AJ*FJ)

      NICE=(LWSTH -HT)/DELHT
      IF ((LWSTHT-HT)/DELHT - NICE .NE. 0.0) RETURN
      IF (HT.NE.LWSTHT) GO TO 605
      DO 600 N=1,2
      S NORM(1,N) = (Q*A3+A4)/CDSQRT(AJ*FJ) /SPVEC(1,N)
      S NORM(2,N) =   (-AJ)  /CDSQRT(AJ*FJ) /SPVEC(2,N)
      S NORM(3,N) =   (Q*AJ) /CDSQRT(AJ*FJ) /SPVEC(3,N)
  600 S NORM(4,N) = (Q*A5+A6)/CDSQRT(AJ*FJ) /SPVEC(4,N)
  605 DO 610 N=1,2
      PART S(1,N) = (Q*A3+A4)/CDSQRT(AJ*FJ) /S NORM(1,N)
      PART S(2,N) =   (-AJ)  /CDSQRT(AJ*FJ) /S NORM(2,N)
      PART S(3,N) =   (Q*AJ) /CDSQRT(AJ*FJ) /S NORM(3,N)
  610 PART S(4,N) = (Q*A5+A6)/CDSQRT(AJ*FJ) /S NORM(4,N)
      RETURN
      END
```

```
      SUBROUTINE QUARTC (FOURB3, SIXB2, FOURB1, B0, Q)
C
C   QUARTC FINDS THE ROOTS OF A QUARTIC
C   POLYNOMIAL, FROM THE CLOSED FORM.
C
      IMPLICIT REAL *8 (A-H,O-Z)
      COMPLEX*16
     $         B3,B2,B1,B0,Q,FOURB3,SIXB2,FOURB1,B3SQ,H,I,G,HPRIME,GPRIME
     1,SQROOT,PPLUS,P,LOGP,CBERT0,CBERT1,CBERT2,OMEGA1,OMEGA2,ROOTP,ROOT
     $Q,ROOTR
      REAL*8 MGPLUS,MGMNUS
      DIMENSION Q(4),PRI(2)
      EQUIVALENCE (P,PRI)
      DATA OMEGA1/(-5.0D-1, 8.660254038D-1)/
      DATA OMEGA2/(-5.0D-1,-8.660254038D-1)/
      DATA  PRECSN /1.0D-10/
C
C
      B3=FOURB3*0.25
      B2=SIXB2/6.0
      B1=FOURB1*0.25
      B3SQ=B3**2
      H=B2-B3SQ
      I=B0-4.0*B3*B1+3.0*B2**2
      G=B1+B3*(-3.0*B2+2.0*B3SQ)
      HPRIME=-I/12.0
      GPRIME=-G**2/4.0-H*(H**2+3.0*HPRIME)
      SQROOT = CDSQRT (GPRIME**2 + 4.0 * HPRIME**3)
      P=(-GPRIME+SQROOT)*0.5
      MGPLUS=DABS(PRI(1))+DABS(PRI(2))
      PPLUS=P
      P=(-GPRIME-SQROOT)*0.5
      MGMNUS=DABS(PRI(1))+DABS(PRI(2))
      IF (MGPLUS.GT.MGMNUS) P=PPLUS
      LOGP = CDLOG (P)
      CBERT0 = CDEXP (LOGP / 3.0)
      CBERT1=OMEGA1*CBERT0
      CBERT2=OMEGA2*CBERT0
      ROOTP = CDSQRT (CBERT0 - HPRIME / CBERT0 - H)
      ROOTQ = CDSQRT (CBERT1 - HPRIME / CBERT1 - H)
      ROOTR = CDSQRT (CBERT2 - HPRIME / CBERT2 - H)
      IF (CDABS(G) .LE. 1.0D-50) GO TO 5
      SIGN=-ROOTP*ROOTQ*ROOTR*2.0/G
      IF (SIGN.LT.0.0) ROOTR=-ROOTR
    5 Q(1)=+ROOTP+ROOTQ+ROOTR-B3
      Q(2)=+ROOTP-ROOTQ-ROOTR-B3
      Q(3)=-ROOTP+ROOTQ-ROOTR-B3
      Q(4)=-ROOTP-ROOTQ+ROOTR-B3
      DO 20 N=1,4
      ITER = 0
  200 CONTINUE
      ROOTP = Q(N)**4 + FOURB3 * Q(N)**3 + SIXB2 * Q(N)**2
     $ + FOURB1 * Q(N) + B0
      ROOTQ = 4.0 * Q(N)**3 + 3.0 * FOURB3 * Q(N)**2
```

```
     $ + 2.0 * SIXB2 * Q(N) + FOURB1
      ROOTR = ROOTP / ROOTQ
      Q(N) = Q(N) - ROOTR
      IF (CDABS (ROOTR).LT.PRECSN)  GO TO 20
      ITER = ITER + 1
      IF (ITER.LT.10)  GO TO 200
      PRINT 900, ITER, Q(N)
   20 CONTINUE
      RETURN
  900 FORMAT (I3,' ITERATIONS, Q = ',
     $E15.5,E13.5,' FAILS TO CONVERGE')
      END
```

```
      SUBROUTINE WF BNDY(B)
C
C  WF BNDY COMPUTES THE VECTOR B,
C  WHICH DETERMINES HOW TO COMBINE
C  THE SOLUTION VECTORS IN ORDER TO
C  SATISFY THE BOUNDARY CONDITIONS.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/T MTX/ T11,T31,T41,T12,T32,T42,T14,T34,T44
      COMMON/M MTX/M
      COMMON/CS/COSINE,   SINE,IGNOR2(8)
      COMMON/WF CON/IGNOR3(2),WAVENR
      COMMON/P MTX/P,IGNOR4(32)
      COMMON/EXC IN/XMTRHT,RXHT
      COMMON/WF FLAG/ ISKIP(3),IDBG
      COMMON/EXTRA/XTRA,R(4)
      COMMON/WFINPT/THETA,FREQ,AZIM,CODIP,MAGFLD,COEFNU(5),EXPNU(5),
     $             TOPHT,LWSTHT,WKBHT,DELHT,H,ALPHA,SIGMA,EPSLON
      COMMON/S MTX/PP(4,4)
      COMPLEX*16 T(4,4),T11,T31,T41,T12,T32,T42,T14,T34,T44,
     $          Q(4),SPVEC(4,4),ASPVEC(4,4),BETA(2,2),P(4,2),
     $          BB(4,4,2),TEMP,I/(0.D0,1.D0)/,M(3,3),SOURCE(9,2),SINE,
     $          BP(4,2),AAA(4),BBB(4),B(12)
      COMPLEX*16 TN(4,4,4),SUM(4,4,4),PROD(4,4),SPVECN(4,4)
      COMPLEX*16 R,RBAR11,RBAR22,XTRA,COSINE,THETA,SMALL F,PP,QQ
      REAL*8 MAGFLD,LWSTHT
      REAL*4 ERR
      DATA QMAX/200.D0/

      CALL TMTRX(0.D0)
      T(1,1) = T11
      T(2,1) = 0.0
      T(3,1) = T31
      T(4,1) = T41
      T(1,2) = T12
      T(2,2) = 0.0
      T(3,2) = T32
      T(4,2) = T42
      T(1,3) = 0.0
      T(2,3) = 1.0
      T(3,3) = 0.0
      T(4,3) = 0.0
      T(1,4) = T14
      T(2,4) = 0.0
      T(3,4) = T34
      T(4,4) = T44
      CALL EIGVAL(T,Q)
      DO 10 N=1,4
      QT = CDABS(Q(N))
      IF(QT .GT. QMAX) IFAIL=1
 10   CONTINUE
      CALL EIGVEC(T,Q,SPVEC)
      CALL WFSORT(Q,SPVEC,IFAIL)
C
```

```
      DO 30 N=1,2
      BB(1,1,N) = Q(N)**2*(T(4,4)-Q(N))-T(2,3)*(T(3,2)*(T(4,4)-Q(N))
     $ -T(4,2)*T(3,4))
      BB(1,2,N) = T(1,2)*Q(N)*(T(4,4)-Q(N))-Q(N)*T(1,4)*T(4,2)
      BB(1,3,N) = T(1,2)*T(2,3)*(T(4,4)-Q(N))-T(1,4)*T(4,2)*T(2,3)
      BB(1,4,N) = -T(1,2)*T(2,3)*T(3,4)-T(1,4)*(Q(N)**2-T(3,2)*T(2,3))
      BB(2,1,N) = T(2,3)*(T(3,1)*(T(4,4)-Q(N))-T(4,1)*T(3,4))
      BB(2,2,N) = -Q(N)*(T(1,1)-Q(N))*(T(4,4)-Q(N))+T(1,4)*T(4,1)*Q(N)
      BB(2,3,N) = -T(2,3)*(T(1,1)-Q(N))*(T(4,4)-Q(N))+T(1,4)*T(4,1)
     $ *T(2,3)
      BB(2,4,N) = T(2,3)*T(3,4)*(T(1,1)-Q(N))-T(1,4)*T(3,1)*T(2,3)
      BB(3,1,N) = Q(N)*T(3,1)*(T(4,4)-Q(N))-Q(N)*T(4,1)*T(3,4)
      BB(3,2,N) = -(T(1,1)-Q(N))*(T(3,2)*(T(4,4)-Q(N))-T(4,2)*T(3,4))
     $ +T(1,2)*(T(3,1)*(T(4,4)-Q(N))-T(4,1)*T(3,4))
     $ -T(1,4)*(T(3,1)*T(4,2)-T(4,1)*T(3,2))
      BB(3,3,N) = -Q(N)*(T(1,1)-Q(N))*(T(4,4)-Q(N))+Q(N)*T(1,4)*T(4,1)
      BB(3,4,N) = Q(N)*T(3,4)*(T(1,1)-Q(N))-Q(N)*T(1,4)*T(3,1)
      BB(4,1,N) = -Q(N)**2*T(4,1)-T(2,3)*(T(3,1)*T(4,2)-T(4,1)*T(3,2))
      BB(4,2,N) = Q(N)*(T(1,1)-Q(N))*T(4,2)-T(1,2)*Q(N)*T(4,1)
      BB(4,3,N) = T(4,2)*T(2,3)*(T(1,1)-Q(N))-T(1,2)*T(4,1)*T(2,3)
   30 BB(4,4,N) = (T(1,1)-Q(N))*(Q(N)**2-T(3,2)*T(2,3))+T(1,2)*T(3,1)
     $ *T(2,3)
      DO 16 N=1,4
      TN(1,1,N) = T(1,1)-Q(N)
      TN(2,2,N) = T(2,2)-Q(N)
      TN(3,3,N) = T(3,3)-Q(N)
      TN(4,4,N) = T(4,4)-Q(N)
      TN(1,2,N) = T(1,2)
      TN(1,3,N) = T(1,3)
      TN(1,4,N) = T(1,4)
      TN(2,1,N) = T(2,1)
      TN(2,3,N) = T(2,3)
      TN(2,4,N) = T(2,4)
      TN(3,1,N) = T(3,1)
      TN(3,2,N) = T(3,2)
      TN(3,4,N) = T(3,4)
      TN(4.1,N) = T(4,1)
      TN(4,2,N) = T(4,2)
   16 TN(4,3,N) = T(4,3)
      DO 17 J=1,4
      DO 17 K=1,4
      DO 17 N=1,2
      SUM(J,K,N) = TN(J,1,N)*BB(1,K,N)+TN(J,2,N)*BB(2,K,N)
     $ +TN(J,3,N)*BB(3,K,N)+TN(J,4,N)*BB(4,K,N)
   17 CONTINUE
      DO 40 N=1,2
      TEMP = CDEXP(I*WAVENR*XMTRHT*Q(N))/(1.+M(3,3))
      SOURCE(1,N)    = -TEMP*SINE**2
      SOURCE(2,N) = 0.0
      SOURCE(3,N) = TEMP*M(2,3)*SINE
      SOURCE(4,N) = -TEMP*M(1,3)*SINE
      TEMP = (1.+M(3,3))*TEMP
      SOURCE(5,N) = -TEMP*SINE
      SOURCE(6,N) = -SOURCE(5,N)
      SOURCE(7,N) =-SOURCE(5,N)*SINE
      SOURCE(8,N) = SOURCE(5,N)
   40 SOURCE(9,N) = SOURCE(5,N)
```

```
      IF (WRGHT.LT.LWSTHT) GO TO  500
      DO 50 II=1,4
      BP(II,1) = (BB(II,1,1)*SOURCE(1,1)+BB(II,2,1)*SOURCE(2,1)
     $    +BB(II,3,1)*SOURCE(3,1)+BB(II,4,1)*SOURCE(4,1))/
     $    ((Q(1)-Q(2))*(Q(1)-Q(3))*(Q(1)-Q(4)))
 50   BP(II,2) = (BB(II,1,2)*SOURCE(1,2)+BB(II,2,2)*SOURCE(2,2)
     $    +BB(II,3,2)*SOURCE(3,2)+BB(II,4,2)*SOURCE(4,2))/
     $    ((Q(2)-Q(1))*(Q(2)-Q(3))*(Q(2)-Q(4)))
      DO 60 II=1,4
      AAA(II) =(BP(II,1)/(SPVEC(II,1)*BETA(2,1))-BP(II,2)/(SPVEC(II,2)*
     $ BETA(2,2)))/(BETA(1,1)/BETA(2,1)-BETA(1,2)/BETA(2,2))
      BBB(II) = (BP(II,1)/SPVEC(II,1)-AAA(II)*BETA(1,1))/
     $ BETA(2,1)
 60   CONTINUE
      B(1) = AAA(1)
      B(2) = BBB(1)
      BP(1,1) = SOURCE(5,1)*BB(1,3,1)
     $  /((Q(1)-Q(2))*(Q(1)-Q(3))*(Q(1)-Q(4)))
      BP(1,2) = SOURCE(5,2)*BB(1,3,2)
     $  /((Q(2)-Q(1))*(Q(2)-Q(3))*(Q(2)-Q(4)))
      B(3) = (BP(1,1)/(SPVEC(1,1)*BETA(2,1))-BP(1,2)/(SPVEC(1,2)*
     $        BETA(2,2)))/(BETA(1,1)/BETA(2,1)-BETA(1,2)/BETA(2,2))
      B(4) = (BP(1,1)/SPVEC(1,1)-B(3)*BETA(1,1))/BETA(2,1)
      BP(1,1) = SOURCE(6,1)*BB(1,4,1)
     $  /((Q(1)-Q(2))*(Q(1)-Q(3))*(Q(1)-Q(4)))
      BP(1,2) = SOURCE(6,2)*BB(1,4,2)
     $  /((Q(2)-Q(1))*(Q(2)-Q(3))*(Q(2)-Q(4)))
      B(5) = (BP(1,1)/(SPVEC(1,1)*BETA(2,1))-BP(1,2)/(SPVEC(1,2)*
     $        BETA(2,2)))/(BETA(1,1)/BETA(2,1)-BETA(1,2)/BETA(2,2))
      B(6) = (BP(1,1)/SPVEC(1,1)-B(5)*BETA(1,1))/BETA(2,1)
      BP(1,1) = SOURCE(7,1)*BB(1,3,1)
     $  /((Q(1)-Q(2))*(Q(1)-Q(3))*(Q(1)-Q(4)))
      BP(1,2) = SOURCE(7,2)*BB(1,3,2)
     $  /((Q(2)-Q(1))*(Q(2)-Q(3))*(Q(2)-Q(4)))
      B(7) = (BP(1,1)/(SPVEC(1,1)*BETA(2,1))-BP(1,2)/(SPVEC(1,2)*
     $        BETA(2,2)))/(BETA(1,1)/BETA(2,1)-BETA(1,2)/BETA(2,2))
      B(8) = (BP(1,1)/SPVEC(1,1)-B(7)*BETA(1,1))/BETA(2,1)
      BP(1,1) = SOURCE(8,1)*BB(1,1,1)
     $  /((Q(1)-Q(2))*(Q(1)-Q(3))*(Q(1)-Q(4)))
      BP(1,2) = SOURCE(8,2)*BB(1,1,2)
     $  /((Q(2)-Q(1))*(Q(2)-Q(3))*(Q(2)-Q(4)))
      B(9) = (BP(1,1)/(SPVEC(1,1)*BETA(2,1))-BP(1,2)/(SPVEC(1,2)*
     $        BETA(2,2)))/(BETA(1,1)/BETA(2,1)-BETA(1,2)/BETA(2,2))
      B(10)= (BP(1,1)/SPVEC(1,1)-B(9)*BETA(1,1))/BETA(2,1)
      BP(1,1) = SOURCE(9,1)*BB(1,2,1)
     $  /((Q(1)-Q(2))*(Q(1)-Q(3))*(Q(1)-Q(4)))
      BP(1,2) = SOURCE(9,2)*BB(1,2,2)
     $  /((Q(2)-Q(1))*(Q(2)-Q(3))*(Q(2)-Q(4)))
      B(11)= (BP(1,1)/(SPVEC(1,1)*BETA(2,1))-BP(1,2)/(SPVEC(1,2)*
     $        BETA(2,2)))/(BETA(1,1)/BETA(2,1)-BETA(1,2)/BETA(2,2))
      B(12)= (BP(1,1)/SPVEC(1,1)-B(11)*BETA(1,1))/BETA(2,1)
      RETURN
C
  500 II=4
      QC = (Q(2)-Q(1))*(Q(2)-Q(3))*(Q(2)-Q(4))
      BP(II,2) = (BB(II,1,2)*SOURCE(1,2)+BB(II,2,2)*SOURCE(2,2)
     $    +BB(II,3,2)*SOURCE(3,2)+BB(II,4,2)*SOURCE(4,2))
```

```
      B(1)=BP(II,2)/(QQ*P(II,2))
      B(2)=SMALLF*B(1)
      B(3)=SOURCE(5,2)*BB(II,3,2)/(QQ*P(II,2))
      B(4)=SMALLF*B(3)
      B(5)=SOURCE(6,2)*BB(II,4,2)/(QQ*P(II,2))
      B(6)=SMALLF*B(5)
      B(7)=SOURCE(7,2)*BB(II,3,2)/(QQ*P(II,2))
      B(8)=SMALLF*B(7)
      B(9)=SOURCE(8,2)*BB(II,1,2)/(QQ*P(II,2))
      B(10)=SMALLF*B(9)
      B(11)=SOURCE(9,2)*BB(II,2,2)/(QQ*P(II,2))
      B(12)=SMALLF*B(11)
      RETURN

      ENTRY DECOMP(HT)
      CALL TMTRX(HT)
      T(1,1) = T11
      T(3,1) = T31
      T(4,1) = T41
      T(1,2) = T12
      T(3,2) = T32
      T(4,2) = T42
      T(1,4) = T14
      T(3,4) = T34
      T(4,4) = T44
      T(2,1) = 0.0
      T(2,2) = 0.0
      T(1,3) = 0.0
      T(2,3) = 1.0
      T(3,3) = 0.0
      T(4,3) = 0.0
      T(2,4) = 0.0
      CALL EIGVAL(T,Q)
      DO 101N=1,4
      QT = CDABS(Q(N))
      IF(QT .GT. QMAX) IFAIL=1
 10 1 CONTINUE
      CALL EIGVEC(T,Q,SPVEC)
      CALL WFSORT(Q,SPVEC,IFAIL)
      DO 311 J=1,4
      DO 311 K=1,4
 311  SPVECN(J,K) = SPVEC(J,K)
      CALL CINVER(SPVECN,ASPVEC,4,4,ERR)
      DO 15 J=1,4
      DO 15 K=1,4
      PROD(J,K) = ASPVEC(J,1)*SPVEC(1,K)+ASPVEC(J,2)*SPVEC(2,K)
     $          +ASPVEC(J,3)*SPVEC(3,K)+ASPVEC(J,4)*SPVEC(4,K)
 15   CONTINUE
      DO 20 N=1,2
      DO 20 NN=1,2
      BETA(N,NN) = ASPVEC(NN,1)*P(1,N)+ASPVEC(NN,2)*P(2,N)
     $  +ASPVEC(NN,3)*P(3,N)+ASPVEC(NN,4)*P(4,N)
 20   CONTINUE
      CALL RBARS(COSINE,SINE,RBAR11,RBAR22)
      IF (CDABS(1.-R(1)*RBAR11) .LT. CDABS(1.-R(2)*RBAR22)) GO TO 230
      SMALL F = -(1.-R(1)*RBAR11)/(R(3)*R(2))
      GO TO 240
```

```
230 SMALL F = -R(4)*R(1)/(1.-R(2)*RBAR22)
240 CONTINUE
    DO 507 II=1,4
    DO 507 N=1,2
507 PP(II,N) = (BETA(1,N)+SMALL F*BETA(2,N))*SPVEC(II,N)
    RETURN
    END
```

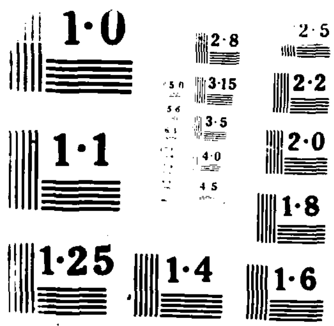
```
      SUBROUTINE EIGVAL (A, SIG)
      IMPLICIT  REAL*8 (A-H,O-Z)
      COMPLEX*16  A, SIG
      DIMENSION  A(4,4), SIG(4)
C
C   COMPUTES 4 EIGENVALUES OF MATRIX A,
C   AND RETURNS THEM IN SIG.
C   THE COEFFICIENTS OF THE CHARACTERISTIC
C   POLYNOMIAL OF A ARE CALCULATED, AND
C   IT ROOTS COMPUTED BY QUARTC, A QUARTIC
C   POLYNOMIAL ROUTINE.
C
      COMMON/WF FLAG/PRECSN,ISO,IDBG
      COMPLEX*16  B(4), BX(3), BY(3)
      COMPLEX*16  B0, B1, SQROOT, SIGN
      DIMENSION  INDEX(4,4)
      DATA  INDEX /0, 4, 4, 3,
     $ 3, 0, 4, 3, 2, 1, 0, 2, 2, 1, 1, 0/
C
C
      IF(ISO .NE. 0) GO TO 600
      DO 100  I = 1,4
  100 B(I) = 0.0
      DO 500  I = 1,3
      IP = I + 1
      DO 500  J = IP,4
      II = INDEX(I,J)
      JJ = INDEX(J,I)
      BX(1) = A(I,1) * A(J,2) - A(I,2) * A(J,1)
      BX(2) = 0.0
      BX(3) = 0.0
      IF (J.NE.2)  GO TO 200
      BX(2) = - A(1,1) - A(2,2)
      BX(3) = 1.0
      GO TO 250
  200 IF (I.EQ.1)  BX(2) = - A(J,2)
      IF (I.EQ.2)  BX(2) = A(J,1)
C
  250 BY(1) = A(II,3) * A(JJ,4) - A(II,4) * A(JJ,3)
      BY(2) = 0.0
      BY(3) = 0.0
      IF (II.NE.3)  GO TO 300
      BY(2) = - A(3,3) - A(4,4)
      BY(3) = 1.0
      GO TO 350
  300 IF (JJ.EQ.4) BY(2) = - A(II,3)
      IF (JJ.EQ.3)  BY(2) = A(II,4)
C
  350 SIGN = (-1)**(I+J-1)
      B(1) = B(1) + SIGN * BX(1) * BY(1)
      B(2) = B(2) + SIGN *
     $ (BX(2) * BY(1) + BX(1) * BY(2))
      B(3) = B(3) + SIGN *
     $ (BX(3) * BY(1) + BX(2) * BY(2) + BX(1) * BY(3))
```

```
      B(4) = B(4) + SIGN *
     $ (BX(2) * BY(3) + BX(3) * BY(2))
  500 CONTINUE
C
      CALL QUARTC (B(4), B(3), B(2), B(1), SIG)
      RETURN
C
C
C   COMPUTES THE EIGENVALUES      X 0 0 X
C   OF A AS THE ROOTS OF TWO      0 X X 0
C   QUADRATICS IF A HAS THE       0 X X 0
C   SPECIAL FORM ON THE RIGHT.    X 0 0 X
  600 CONTINUE
      B1 = 0.5 * (A(1,1) + A(4,4))
      B0 = A(1,1) * A(4,4) - A(1,4) * A(4,1)
      SQROOT = CDSQRT (B1**2 - B0)
      SIG(1) = B1 + SQROOT
      SIG(4) = B1 - SQROOT
      B1 = 0.5 * (A(2,2) +A(3,3))
      B0 = A(2,2) * A(3,3) - A(2,3) * A(3,2)
      SQROOT = CDSQRT (B1**2 - B0)
      SIG(2) = B1 + SQROOT
      SIG(3) = B1 - SQROOT
      RETURN
      END
```

```
      SUBROUTINE EIGVEC (A, SIG, VEC)
      IMPLICIT  REAL*8 (A-H,O-Z)
      DIMENSION  A(4,4), SIG(4), VEC(4,4)
      COMPLEX*16  A, SIG, VEC
C
C   COMPUTES FOUR EIGENVECTORS OF A,
C   AND RETURNS THEM IN VEC.
C   VECTORS ARE COMPUTED AS SOLUTION OF
C   REDUCED SYSTEM OF LINEAR EQNS.
C
      COMMON/WF FLAG/PRECSN,ISO,IDBG
      DIMENSION  B(3,3), Y(3), X(3)
      COMPLEX*16  B, Y, X
      REAL*4  ERR
C
C
      IF(ISO .NE. 0) GO TO 600
C   COMPUTE VECTORS WITH VEC(1,J) = A(1,4).
C   (THIS IS DONE AGREE WITH SPECIAL
C    ISOTROPIC CASE COMPUTED BELOW.)
      DO 100  J = 1,4,3
      B(1,1) = A(2,2) - SIG(J)
      B(1,2) = A(2,3)
      B(1,3) = A(2,4)
      Y(1) = - A(2,1) * A(1,4)
      B(2,1) = A(3,2)
      B(2,2) = A(3,3) - SIG(J)
      B(2,3) = A(3,4)
      Y(2) = - A(3,1) * A(1,4)
      B(3,1) = A(4,2)
      B(3,2) = A(4,3)
      B(3,3) = A(4,4) - SIG(J)
      Y(3) = - A(4,1) * A(1,4)
      CALL CLINEQ (B, Y, X, 3, 3, 0, ERR)
      VEC(1,J) = A(1,4)
      VEC(2,J) = X(1)
      VEC(3,J) = X(2)
      VEC(4,J) = X(3)
  100 CONTINUE
C
C   COMPUTE VECTORS WITH VEC(2,J) = A(2,3)
      DO 200  J = 2,3
      B(1,1) = A(1,1) - SIG(J)
      B(1,2) = A(1,3)
      B(1,3) = A(1,4)
      Y(1) = - A(1,2) * A(2,3)
      B(2,1) = A(3,1)
      B(2,2) = A(3,3) - SIG(J)
      B(2,3) = A(3,4)
      Y(2) = - A(3,2) * A(2,3)
      B(3,1) = A(4,1)
      B(3,2) = A(4,3)
      B(3,3) = A(4,4) - SIG(J)
      Y(3) = - A(4,2) * A(2,3)
```

AD-A169 006 A WKB PROGRAM FOR ELF/VLF EARTH-IONOSPHERE EXCITATION BY SOURCES AT SATELLITE HEIGHTS(U) NAVAL OCEAN SYSTEMS CENTER SAN DIEGO CA R A PAPPERT ET AL. SEP 82 2/2

UNCLASSIFIED NOSC/TR-816 F/G 20/4 NL


1·0
1·1
1·25
1·4
1·6
1·8
2·0
2·2
2·5
2·8
3·15
3·5
4·0
4·5

```
      CALL CLINEQ (B, Y, X, 3, 3, 0, ERR)
      VEC(1,J) = X(1)
      VEC(2,J) = A(2,3)
      VEC(3,J) = X(2)
      VEC(4,J) = X(3)
  200 CONTINUE
      RETURN
C
C
C   COMPUTE EIGENVECTORS IN        X 0 0 X
C   SIMPLIFIED FORM IF A HAS       0 X X 0
C   THE SPECIAL FORM ON            0 X X 0
C   THE RIGHT.                     X 0 0 X
C
  600 CONTINUE
C   COMPUTE VECTORS WITH VEC(1,J) = A(1,4).
      DO 300  J = 1,4,3
      VEC(1,J) = A(1,4)
      VEC(2,J) = 0.0
      VEC(3,J) = 0.0
      VEC(4,J) = SIG(J) - A(1,1)
  300 CONTINUE
C
C   COMPUTE VECTORS WITH VEC(2,J) = A(2,3)
      DO 400 J = 2,3
      VEC(1,J) = 0.0
      VEC(2,J) = A(2,3)
      VEC(3,J) = SIG(J) - A(2,2)
      VEC(4,J) = 0.0
  400 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE WF SORT (Q, A, IFAIL)
      IMPLICIT  REAL*8 (A-H,O-Z)
      COMPLEX*16  Q(4), A(4,4)
C
C    WF SORT INITIALLY ARRANGES THE
C    Q VALUES SO THAT THEY OCCUR IN THE
C    ORDER UPGOING FAST (EVANESCENT),
C    UPGOING SLOW (TRAVELLING), DOWNGOING
C    SLOW, AND DOWNGOING FAST.
C
      COMPLEX*16  QT(4), B(4,4), I, CT
      DIMENSION  QI(4)
      DATA  I /(0.0D0,1.0D0)/
C
C
      DO 510 J = 1,4
      QT(J) = Q(J)
      QI(J) = - I * Q(J)
      DO 510  K = 1,4
  510 B(J,K) = A(J,K)
      DO 530  J = 1,4
      T = QI(1)
      JJ = 1
      DO 520  K = 2,4
      IF(QI(K).GE.T)  GO TO 520
      T = QI(K)
      JJ = K
  520 CONTINUE
      Q(J) = QT(JJ)
      DO 525  K  = 1,4
  525 A(K,J) = B(K,JJ)
      QI(JJ) = 1.0D50
  530 CONTINUE
      QI(1) = - I * Q(2)
      QI(2) = Q(2)
      IF (DABS(QI(1)/QI(2)).LT.0.2)  GO TO 550
      IF(QI(1).GT.0.0)  GO TO 850
      QI(1) = - I * Q(3)
      IF( QI(1).LT.0.0)  GO TO 850
      RETURN
  550 QI(1) = Q(3)
      IF (QI(2).GT.QI(1))  GO TO 560
      CT = Q(2)
      Q(2) = Q(3)
      Q(3) = CT
      QI(3) = QI(2)
      QI(2) = QI(1)
      QI(1) = QI(3)
      DO 555  K = 1,4
      CT = A(K,2)
      A(K,2) = A(K,3)
  555 A(K,3) = CT
  560 IF (QI(2).LT.0.0)  GO TO 850
      IF(QI(1).GT.0.0)  GO TO 850
```

```
      QI(1) = - I * Q(4)
      QI(2) = Q(4)
      IF (QI(1).LT.0.2*DABS(QI(2)))  GO TO 850
      RETURN
C
  850 IFAIL = 1
      PRINT 901
      PRINT 902
      PRINT 900, Q, QT
      RETURN
  900 FORMAT (5X,' Q =',4(E15.5,E13.5),/,
     $   PREV Q =',4(E15.5,E13.5))
  901 FORMAT (' ERROR IN INITIAL Q VALUES')
  902 FORMAT (' Q ELEMENTS CANNOT BE ORDERED')
      END
```

```
      SUBROUTINE RBARS(C,S,RBAR11,RBAR22)
C
      IMPLICIT REAL *8 (A-H,O-Z)
      COMMON/WFINPT/THETA,FREQ,AZMUTH,CODIP,MAGFLD,CEFFNU(5),EXPNU(5),
     $ TOPHT,LWSTHT,WKBHT,DELHT,H,ALPHA,SIGMA,EPSLON
      COMMON/STOR/ STORE
      COMMON/WF CON/OMEGA,K
      COMPLEX *16 THETA,I,NGSQ,C,S,SSQ,SQROOT,RTIORT,IKC,
     $          P0,H10,H20,H1PRM0,H2PRM0,CAPH10,CAPH20,
     $          PD,H1D,H2D,H1PRMD,H2PRMD,CAPH1D,CAPH2D,
     $          PZ,H1Z,H2Z,H1PRMZ,H2PRMZ,
     $          A1ST,A2ND,A3RD,A4TH,A1,A2,A3,A4,
     $          EXD,EXDSQ,EXZ,EXZSQ,
     $          RBAR11,RBAR22,Z1,Z2,
     $          DEN12,DEN34,
     $          EX,EY,EZ,HX,HY,HZ
      REAL *8 K,KVRAOT,KVRATT,N0SQ,NDSQ,NZSQ,MAGFLD,LWSTHT
      EQUIVALENCE(PZ,PD),(H1Z,H1D),(H2Z,H2D),(H1PRMZ,H1PRMD),
     $             (H2PRMZ,H2PRMD),(EXD,EXZ),(EXDSQ,EXZSQ)
      DATA I/(0.0D0,1.0D0)/
      DATA TSTTHM/1.0D1/
      DATA EPSLN0/8.85434D-12/
C
C
      D = TOPHT
      SSQ=S*S
      NGSQ = (EPSLON-I*SIGMA/OMEGA)/EPSLN0
      SQROOT=CDSQRT(NGSQ-SSQ)
      THTIM=I*THETA
      IF(THTIM .GT. TSTTHM) GO TO 10
C
      KVRAOT=DEXP(DLOG(K/ALPHA)/3.0)
      KVRATT=KVRAOT**2
      AVRKOT=1.0/KVRAOT
      AVRKTT=AVRKOT**2*0.5
      N0SQ=1.0-ALPHA*(H-STORE)
      RTIORT=N0SQ/NGSQ*SQROOT
      P0=KVRATT*(N0SQ-SSQ)
      CALL MDHNKL (P0,H10,H20,H1PRM0,H2PRM0)
      CAPH10=H1PRM0+AVRKTT*H10
      CAPH20=H2PRM0+AVRKTT*H20
      A1ST=CAPH20-I*RTIORT*KVRAOT*H20
      A2ND=CAPH10-I*RTIORT*KVRAOT*H10
      A3RD=H2PRM0-I*KVRAOT*SQROOT*H20
      A4TH=H1PRM0-I*KVRAOT*SQROOT*H10
      DEN12 = H20*A2ND-H10*A1ST
      DEN34 = H20*A4TH-H10*A3RD
      IF(D .EQ. 0.0) GO TO 10
C
      NDSQ=1.0-ALPHA*(H-D)
      PD=KVRATT*(NDSQ-SSQ)
      CALL MDHNKL (PD,H1D,H2D,H1PRMD,H2PRMD)
      CAPH1D=H1PRMD+AVRKTT*H1D
      CAPH2D=H2PRMD+AVRKTT*H2D
```

```
C
      A1 = C*NDSQ*(H2D*A2ND-H1D*A1ST)
      A2=I*AVRKOT*(CAPH1D*A1ST-CAPH2D*A2ND)
      A3=I*AVRKOT*(H2PRMD*A4TH-H1PRMD*A3RD)
      A4 = C*(H2D*A4TH-H1D*A3RD)
      REAR11=(A1-A2)/(A1+A2)
      RBAR22=(A3+A4)/(A4-A3)
      RETURN
C
C     FLAT EARTH
   10 IKC = I*K*C
      EXD = CDEXP(-IKC*(STORE-D))
      EXDSQ = EXD**2
      Z1=(NGSQ*C-SQROOT)/(NGSQ*C+SQROOT)
      Z2=(C-SQROOT)/(C+SQROOT)
      RBAR11=Z1*EXDSQ
      RBAR22=Z2*EXDSQ
      RETURN
C
C
      ENTRY HT GAIN(ALT,EX,EY,EZ,HX,HY,HZ)
   40 IF(THTIM .GT. TSTTHM) GO TO 50
      NZSQ = 1.0-ALPHA*(H-ALT)
      PZ = KVRATT*(NZSQ-SSQ)
      CALL MUHNKL(PZ,H1Z,H2Z,H1PRMZ,H2PRMZ)
      EXPON = DEXP(-0.5*ALPHA*ALT)
      HY = (H2Z*A2ND-H1Z*A1ST)*EXPON/DEN12
      EY = (H2Z*A4TH-H1Z*A3RD)/DEN34
      EX =       I*AVRKOT*((H2PRMZ*A2ND-H1PRMZ*A1ST)/DEN12
     $          *EXPON+AVRKTT*HY)/NZSQ
      EZ = -S/NZSQ*HY
      HZ = S*EY
      HX = AVRKOT/I*(H2PRMZ*A4TH-H1PRMZ*A3RD)/DEN34
      RETURN
C
C     FLAT EARTH
50    EXZ = CDEXP(-IKC*(STORE-ALT))
      EXZSQ = EXZ**2
      HY = (1.0+Z1*EXZSQ)/(1.0+Z1)/EXZ
      EY = (1.0+Z2*EXZSQ)/(1.0+Z2)/EXZ
      EX = -C*(1.0-Z1*EXZSQ)/(1.0+Z1)/EXZ
      EZ = -S*HY
      HZ = S*EY
      HX = C*(1.0-Z2*EXZSQ)/(1.0+Z2)/EXZ
      RETURN
C
      END
```

```
      SUBROUTINE MDHNKL (Z,H1,H2,H1PRME,H2PRME)
      IMPLICIT REAL *8 (A-H,O-Z)
      COMPLEX*16 Z,I,H1,H2,H1PRME,H2PRME,ZPOWER,TERM1,TERM2,
     $           ZTERM,TERM,SUM1,SUM2,SUM3,SUM4,SQRTZB,
     $           EXP1,EXP2,EXP3,EXP4,EXP5,GM2F,GPMFP,MPOWER,BETA,RTZ,
     $           CONST1,CONST2,CONST3,CONST4
      DIMENSION A(23), B(23), C(23), D(23), CAP(14)
      DATA A/
     $   9.30436716930000D-01,3.10145572309700D 01,2.06763714873160D 02,
     $   5.74343652425450D 02,8.70217655190080D 02,8.28778719228640D 02,
     $   5.41685437404340D 02,2.57945446383020D 02,9.34584950663100D 01,
     $   2.66263518707400D 01,6.12100043005600D 00,1.15928038448000D 00,
     $   1.84012759441000D-01,2.48330309640000D-02,2.88420801000000D-03,
     $   2.91334142000000D-04,2.58274050000000D-05,2.02568600000000D-06,
     $   1.41557000000000D-07,8.87000000000000D-09,5.01000000000000D-10,
     $   2.60000000000000D-11,1.00000000000000D-12/
      DATA B/
     $   6.78298725140000D-01,1.13049787524000D 01,5.38332321543100D 01,
     $   1.19629404787350D 02,1.53371031778650D 02,1.27809193148880D 02,
     $   7.47422182157200D 01,3.23559386215200D 01,1.07853128738400D 01,
     $   2.85325737403000D 00,6.13603736351000D-01,1.03376780098000D-01,
     $   1.64229399550000D-02,2.10550512200000D-03,2.33167788000000D-04,
     $   2.25282890000000D-05,1.91567100000000D-06,1.44470000000000D-07,
     $   9.72900000000000D-09,5.89000000000000D-10,3.20000000000000D-11,
     $   2.00000000000000D-12,0.00000000000000D 00/
      DATA C/
     $   4.65218358460000D-01,6.20291144619000D 00,2.58454643591500D 01,
     $   5.22130593114000D 01,6.21584039421500D 01,4.87516893663900D 01,
     $   2.70842718702200D 01,1.12150194079600D 01,3.59455750255000D 00,
     $   9.18150064510000D-01,1.91281263439000D-01,3.31222960990000D-02,
     $   4.84244103800000D-03,6.05683682000000D-04,6.55501820000000D-05,
     $   6.19859900000000D-06,5.16550000000000D-07,3.82200000000000D-08,
     $   2.52800000000000D-09,1.50000000000000D-10,8.00000000000000D-12,
     $   0.00000000000000D 00,0.00000000000000D 00/
      DATA D/
     $   6.78298725140000D-01,4.52190150096200D 01,3.76832625080150D 02,
     $   1.19629404787350D 03,1.99382341312250D 03,2.04494709038206D 03,
     $   1.42010214609865D 03,7.11830649673510D 02,2.69632821846030D 02,
     $   7.98912064729000D 01,1.90217158268800D 01,3.71881052333900D 00,
     $   6.07648778323000D-01,8.42202048960000D-02,1.00262148690000D-02,
     $   1.03630127800000D-03,9.38678690000000D-05,7.51243500000000D-06,
     $   5.35074000000000D-07,3.41350000000000D-08,1.96200000000000D-09,
     $   1.02000000000000D-10,5.00000000000000D-12/
      DATA CAP/
     $   1.04166666666667D-01,8.35503472222222D-02,1.28226574556327D-01,
     $   2.91849026464140D-01,8.81627267443758D-01,3.32140828186277D 00,
     $   1.49957629868620D 01,7.89230130115870D 01,4.74451538868000D 02,
     $   3.20749009100000D 03,2.40865496000000D 04,1.98923120000000D 05,
     $   1.79190200000000D 06,1.74843770000000D 07/
C
      DATA I/(0.0D0,1.0D0)/
      DATA ROOT3/1.73205080756888D 00/
      DATA ALPHA/8.53667218838951D-01/
      DATA CONST1/( 2.58819045102522D-01,-9.65925826289067D-01)/
```

```
      DATA CONST2/( 2.588190451025220-01, 9.659258262890670-01)/
      DATA CONST3/(-9.65925826289067D-01, 2.588190451025220-01)/
      DATA CONST4/(-9.65925826289067D-01,-2.588190451025220-01)/
C
      ZPOWER=1.0
      SUM3=0.0
      SUM4=0.0
      ZMAG=CDABS(Z)
      IF(ZMAG .GT. 4.2) GO TO 70
      IF(ZMAG .GE. 3.2) GO TO 10
      N=12
      GO TO 30
   10 IF(ZMAG .GE. 4.1) GO TO 20
      N=15
      GO TO 30
   20 N=23
   30 SUM1=0.
      SUM2=0.
      ZTERM=-Z**3/200.0
      DO 50 M=1,N
      SUM1=SUM1+A(M)*ZPOWER
      SUM2=SUM2+B(M)*ZPOWER
      SUM3=SUM3+C(M)*ZPOWER
      SUM4=SUM4+D(M)*ZPOWER
      ZPOWER=ZPOWER*ZTERM
      IF(CDABS(ZPOWER) .LE. 1.0D-30) GO TO 60
   50 CONTINUE
   60 GM2F=I*(Z*SUM2-2.*SUM1)/ROOT3
      GPMFP=I*(SUM4+2.*Z*Z*SUM3)/ROOT3
      H1=Z*SUM2+GM2F
      H2=H1-2.0*GM2F
      H1PRME=SUM4+GPMFP
      H2PRME=H1PRME-2.0*GPMFP
      RETURN
C
   70 SUM1=1.0
      SUM2=1.0
      RTZ=CDSQRT(Z)
      SQRTZB=RTZ*Z
      ZTERM=1/SQRTZB
      MPOWER=1.0
      TERM=-1.5/Z
      DO 80 M=1,14
      ZPOWER=ZPOWER*ZTERM
      MPOWER=MPOWER*(-ZTERM)
      TERM1=CAP(M)*ZPOWER
      TERM2=CAP(M)*MPOWER
      SUM1=SUM1+TERM1
      SUM2=SUM2+TERM2
      SUM3=SUM3+M*TERM1
      SUM4=SUM4+M*TERM2
   80 CONTINUE
      SUM3=SUM3*TERM
      SUM4=SUM4*TERM
      EXP1=CDEXP(2.*I*SQRTZB/3.)
      EXP2=EXP1*CONST1
      EXP3=CONST2/EXP1
```

```
      EXP4=CONST3*EXP1
      EXP5=CONST4/EXP1
      BETA=ALPHA/CDSQRT(RTZ)
      ZREAL=Z
      ZIMAG=-I*Z
      IF (ZREAL.GE.0.0.OR.ZIMAG.GE.0.0)GO TO 90
      H1=BETA*(EXP2*SUM2+EXP5*SUM1)
      H1PRME=BETA*(EXP2*(SUM2*(-0.25/Z+I*RTZ)+SUM4)+EXP5*(SUM1*(-0.25/Z
     $       -I*RTZ)+SUM3))
      GO TO 110
 90   H1=BETA*EXP2*SUM2
      H1PRME=BETA*EXP2*(SUM2*(-0.25/Z+I*RTZ)+SUM4)
 110  IF (ZREAL.GE.0.0.OR.ZIMAG.LT.0.0)GO TO 120
      H2=BETA*(EXP3*SUM1+EXP4*SUM2)
      H2PRME=BETA*(EXP3*(SUM1*(-0.25/Z-I*RTZ)+SUM3)+EXP4*(SUM2*(-0.25/Z
     $       +I*RTZ)+SUM4))
      RETURN
 120  H2=BETA*EXP3*SUM1
      H2PRME=BETA*EXP3*(SUM1*(-0.25/Z-I*RTZ)+SUM3)
      RETURN
      END
```

```
      SUBROUTINE CLIN EQ (A, B, X, N,
     $    N DIM, IFLAG, ERR)
C
C   CLIN EQ USES L-U DECOMPOSITION TO
C   FIND THE TRIANGULAR MATRICES L, U
C   SUCH THAT  L * U = A.  L AND U ARE
C   STORED IN A.  THIS FORM IS USED WITH
C   BACK-SUBSTITUTION TO FIND THE SOLN
C   X OF   A * X = L * U * X = B.
C   N IS THE NUMBER OF EQUATIONS AND
C   N DIM IS THE DIMENSION OF ALL ARRAYS
C   IN THE PARAMETER LIST.
C
C   IF IFLAG = 0, L, U, AND X ARE COMPUTED
C   IF IFLAG IS NON-ZERO, IT IS ASSUMED
C   THAT L AND U HAVE BEEN COMPUTED IN
C   A PREVIOUS CALL AND ARE STILL STORED
C   IN A.  THUS ONLY X IS COMPUTED.
C   ERR  IS THE ESTIMATED RELATIVE
C   ERROR OF THE SOLUTION VECTOR.
C
      COMPLEX*16  A, B, X, T
      INTEGER*4  IROW
      DIMENSION  A(N DIM, N DIM),
     $  B(N DIM), X(N DIM)
      DIMENSION  IROW(50), Q(50)
      DATA  EPS /1.0E-15/
C
      IF (N.GT.50)  GO TO 900
      IF (IFLAG.NE.0)  GO TO 600
      DO 050  I = 1,N
      Q(I) = 0.0
      DO 040  J = 1,N
      QQ = CDABS (A(I,J))
  040 IF (Q(I).LT.QQ)  Q(I) = QQ
      IF (Q(I).EQ.0.0)  GO TO 901
  050 CONTINUE
      ERR = EPS
      PPIV = 0.0
      DO 100  I = 1,N
  100 IROW(I) = I
C
      DO 500  L = 1,N
      PIVOT = 0.0
      K = L - 1
      DO 240  I = L,N
      IF (K.LT.1)  GO TO 230
      DO 220  J = 1,K
  220 A(I,L) = A(I,L) - A(J,L) * A(I,J)
  230 F = CDABS (A(I,L)) / Q(I)
      IF (PIVOT.GT.F)  GO TO 240
      PIVOT = F
      NPIVOT = I
  240 CONTINUE
```

```
      IF (PIVOT.EQ.0.0)  GO TO 901
      IF (PPIV.LE.PIVOT)  GO TO 250
      ERR = ERR * PPIV / PIVOT
      IF (ERR.GE.1.0)  GO TO 901
  250 PPIV = PIVOT
      IF (NPIVOT.EQ.L)  GO TO 280
      Q(NPIVOT) = Q(L)
      J = IROW(L)
      IROW(L) = IROW(NPIVOT)
      IROW(NPIVOT) = J
      DO 260  I = 1,N
      T = A(L,I)
      A(L,I) = A(NPIVOT,I)
      A(NPIVOT,I) = T
  260 CONTINUE
  280 IF (L.EQ.N)  GO TO 500
      T = (1.0D0,0.0D0) / A(L,L)
      K = L + 1
      M = L - 1
      DO 450  I = K,N
      IF (M.LT.1)  GO TO 400
      DO 350  J = 1,M
  350 A(L,I) = A(L,I) - A(L,J) * A(J,I)
  400 A(L,I) = T * A(L,I)
  450 CONTINUE
  500 CONTINUE
      IF (ERR.GT.1.0E-5)   PRINT 998, ERR
C
  600 DO 620  I = 2,N
  620 X(I) = (0.0D0,0.0D0)
      J = IROW(1)
      X(1) = B(J) / A(1,1)
      DO 700  I = 2,N
      J = IROW(I)
      K = I - 1
      DO 650  L = 1,K
  650 X(I) = X(I) + A(I,L) * X(L)
      X(I) = (B(J) - X(I)) / A(I,I)
  700 CONTINUE
      K = N - 1
      DO 800  I = 1,K
      J = N - I
      M = J + 1
      DO 800  L = M,N
      X(J) = X(J) - X(L) * A(J,L)
  800 CONTINUE
      RETURN
C
  900 PRINT 999
      ERR = 1.0
      RETURN
  901 PRINT 997
      ERR = 1.0
      RETURN
  997 FORMAT ('1ERROR IN CLIN EQ, MATRIX IS SINGULAR')
  998 FORMAT (' CAUTION-',
     $ ' CLIN EQ HAS DECOMPOSED AN ILL-CONDITIONED MATRIX.',/,
```

```
     $ ' RESULTS WILL HAVE RELATIVE ERROR =',E11.2)
 999 FORMAT ('1ERROR IN CLIN EQ, MATRIX SIZE GREATER THAN 50')
     END
```

```
      SUBROUTINE CINVER (A, A INV, N, N DIM, ERR)
C
C   CINVER SUCCESSIVELY COMPUTES EACH COLUMN
C   OF THE INVERSE MATRIX OF A, USING THE
C   SUBROUTINE CLIN EQ TO SOLVE A * X = E
C   FOR EACH UNIT VECTOR E.
C   N IS THE ORDER OF THE MATRIX A AND
C   N DIM IS THE DIMENSION OF ALL ARRAYS
C   IN THE PARAMETER LIST.
C   ERR IS THE ESTIMATED RELATIVE ERROR
C   OF THE INVERTED MATRIX.
C   THE INVERSE MATRIX IS
C   RETURNED IN A INV.
C
      COMPLEX*16  A, A INV, B, X
      DIMENSION  A(N DIM, N DIM),
     $ A INV(N DIM, N DIM)
      DIMENSION  B(50), X(50)
C
C
      IF (N.GT.50)  GO TO 900
      II = 0
      DO 100  I = 1,N
  100 B(I) = (0.0D0,0.0D0)
      DO 300  I = 1,N
      B(I) = (1.0D0,0.0D0)
      CALL CLIN EQ (A, B, X, N, N DIM, II, ERR)
      IF (ERR.GE.1.0)  GO TO 901
      DO 200  J = 1,N
  200 A INV(J,I) = X(J)
      B(I) = (0.0D0,0.0D0)
      II = 1
  300 CONTINUE
      IF (ERR.GT.1.0E-5)  GO TO 901
      RETURN
C
  900 PRINT 998
      ERR = 1.0
      RETURN
  901 PRINT 999
      RETURN
  998 FORMAT ('1ERROR IN CINVER, MATRIX SIZE GREATER THAN 50')
  999 FORMAT (' CLIN EQ WAS CALLED BY CINVER')
      END
```

```
      SUBROUTINE MA3(Z1,Z2,Z3,OUT)
      COMPLEX*16 Z1,Z2,Z3
      REAL*8 CDANG
      DIMENSION OUT(6)
      OUT(1)=CDABS(Z1)
      OUT(3)=CDABS(Z2)
      OUT(5)=CDABS(Z3)
      OUT(2)=CDANG(Z1)
      OUT(4)=CDANG(Z2)
      OUT(6)=CDANG(Z3)
      RETURN
      END
```

```
      FUNCTION CDANG(ARG)
C
      IMPLICIT REAL*8 (A-H,O-Z)
      COMPLEX*16 ARG,ARG PRT
      DIMENSION PARTS(2)
      EQUIVALENCE(ARG PRT,PARTS)
C
C
      ARG PRT = ARG
      ARG RL = PARTS(1)
      ARG IM = PARTS(2)
      CDANG = DATAN2(ARG IM,ARG RL)
      RETURN
C
      END
```

```
      BLOCK DATA
C
C   INITIALIZE THE COMMON BLOCK VALUES.
C
      IMPLICIT REAL *8 (A-H,O-Z)
      COMMON/WFINPT/THETA,FREQ,AZMUTH,CODIP,MAGFLD,CEFFNU(5),EXPNU(5),
     $ TOPHT,LWSTHT,WKBHT,DELHT,H,ALPHA,SIGMA,EPSLON
      COMMON/WF/FLAG,PRECSN,ISO,IDBG
      COMMON/WFPROF/ENHT(100),ENLOG(100,5),COLLHT(25),COLLFR(25,5),
     $              LHT,MHT,CHARGE(5),RATIOM(5),NRSPEC
      COMMON/EXC IN/TXHT,RXHT
      REAL*8  MAGFLD, LWSTHT
      COMPLEX*16  THETA
C
C
      DATA MAGFLD /0.0D0/
      DATA CEFFNU/1.816D11,4*0.0D0/
      DATA EXPNU/-1.5D-4,4*0.0D0/
      DATA  TOPHT /100.0D0/,LWSTHT/0.CD0/,WKBHT/000.0D0/
      DATA  H /0.0D0/
      DATA  ALPHA /3.14D-4/
      DATA  SIGMA /4.64D0/
      DATA  EPSLON /7.172015D-10/
      DATA  PRECSN /3.0D-5/
      DATA  IDBG /1/
      DATA CHARGE/-1.0D0,1.0D0,-1.0D0,1.0D0,-1.0D0/
      DATA  RATIOM /1.0D0,4*5.8D4/
      DATA  NRSPEC /1/
      END
```

APPENDIX B

MODE SUM AND PLOTTING PROGRAM LISTING

```
C
C
C     THIS PROGRAM WAS DEVELOPED FOR USE ON THE UNIVAC 1100/82 AT THE
C     NAVAL OCEAN SYSTEMS CENTER.  IT WAS WRITTEN IN ASCII FORTRAN USING
C     THE LEVEL 10R1 COMPILER WHICH CONFORMS TO AMERICAN NATIONAL
C     STANDARDS INSTITUTE FORTRAN-77 STANDARDS.
C     THE DISSPLA PACKAGE WAS USED TO GENERATE THE PLOTS.
C
C     THE PROGRAM USES EXCITAION FACTORS GENERATED BY THE
C     WKB  SATELLITE   PROGRAM TO CALCULATE MODE SUMS FOR THREE
C     ORTHOGONAL FIELD COMPONENTS - EX, EY, AND EZ.  SIX CURVES ARE
C     GENERATED FOR EACH FIELD COMPONENT DESIRED:
C        ELECTRIC DIPOLE - VERTICAL
C        ELECTRIC DIPOLE - HORIZONTAL-BROADSIDE
C        ELECTRIC DIPOLE - HORIZONTAL-ENDON
C        MAGNETIC DIPOLE - VERTICAL
C        MAGNETIC DIPOLE - HORIZONTAL-BROADSIDE
C        MAGNETIC DIPOLE - HORIZONTAL-ENDON
C
C     NOTE THAT THE EXCITATION FACTORS HAVE THE HEIGHT OF THE RECEIVER
C     BUILT INTO THEM SO IF THE USER WISHES MODE SUMS FOR DIFFERENT
C     TRANSMITTER AND/OR RECIEVER HEIGHTS NEW EXCITAION FACTORS MUST
C     BE GENERATED.
C
C
      CALL INPUT
C
      CALL FIELDS
C
      CALL OUTPUT
C
      STOP
      END
```

```
      SUBROUTINE INPUT
C
C
C     THIS SUBROUTINE READS IN AN IDENTIFICATION CARD FOR PLOT LABELS,
C     THE NAMELIST  VARIABLES, AND THE COMPLEX EIGEN ANGLE AND
C     EXCITATION FACTORS FOR EACH MODE.
C
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 IL,IA
      REAL*4 AZIM,CODIP,MAGFLD,SIGMA,EPSR,DIST,AMP
      COMPLEX*16 IM/(0.0D0,1.0D0)/
      COMPLEX*16 THETA,EXCIT
      CHARACTER*14 LABEL
      DIMENSION XTRMAG(3),XTRANG(3)
      COMMON/ONE/RHOMIN,RHOMAX,DELRHO,MODES,IPLOT
      COMMON/TWO/AZIM,CODIP,MAGFLD,SIGMA,EPSR
      COMMON/THREE/THETA(10),EXCIT(3,6,10)
      COMMON/FOUR/FREQ,IL,IA,TXHT,RXHT,DIST(500),AMP(6,500),NPTS,
     $  IDPLOT(20),LABEL
      NAMELIST/DATUM/FREQ,RHOMIN,RHOMAX,IL,IA,MODES,DELRHO,TXHT,RXHT,
     $               IPLOT,AZIM,CODIP,MAGFLD,SIGMA,EPSR
C
C     DESCRIPTION OF NAMELIST VARIABLES
C
C     FREQ   - FREQUENCY IN KHZ
C     RHOMIN - MINIMUM DISTANCE (IN KM) AT WHICH FIELDS ARE PRINTED
C              AND PLOTTED
C     RHOMAX - MAXIMUM DISTANCE (IN KM) AT WHICH FIELDS ARE PRINTED
C              AND PLOTTED
C     IL     - ELECTRIC DIPOLE CURRENT MOMENT IN AMP METERS
C     IA     - MAGNETIC DIPOLE MOMENT IN AMP METERS SQUARED
C     MODES  - NUMBER OF MODES USED IN MODE SUMS
C     DELRHO - DISTANCE INCREMENT (IN KM) AT WHICH FIELDS ARE PRINTED
C              AND PLOTTED
C     TXHT   - TRANSMITTER HEIGHT IN KM
C     RXHT   - RECEIVER HEIGHT IN KM
C     IPLOT  - PLOTTING FLAG
C              IPLOT=0  NO PLOTS
C              IPLOT=1  EZ PLOT
C              IPLOT=2  EY AND EZ PLOTS
C              IPLOT=3  EX, EY, AND EZ PLOTS
C     THE FOLLOWING FIVE VARIABLES ARE USED SOLEY FOR IDENTICATION
C     PURPOSES ON THE PLOTS
C     AZIM   - AZIMUTH IN DEGREES
C     CODIP  - CODIP IN DEGREES
C     MAGFLD - MAGNETIC FIELD IN WEBERS PER SQUARE METER
C     SIGMA  - CONDUCTIVITY IN SIEMENS PER METER
C     EPSR   - DIELECTRIC CONSTANT
C
C
      PRINT 902
      READ 900,IDPLOT
      PRINT 910,IDPLOT
      READ(5,DATUM,END=999)
```

```
      WRITE(6,DATUM)
      DO 50 M=1,MODES
      READ 915,THETA(M)
      PRINT 917,M,THETA(M)
      DO 50 L=1,6
      READ 920,(XTRMAG(J),XTRANG(J),J=1,3)
      DO 50 J=1,3
      EXCIT(J,L,M) = XTRMAG(J)*(DCOS(XTRANG(J))+IM*DSIN(XTRANG(J)))
50    CONTINUE
C
900   FORMAT(20A4)
902   FORMAT('1')
910   FORMAT('0',20A4)
915   FORMAT(8X,F6.3,1X,F7.3)
917   FORMAT(' MODE ',I2,'  THETA = ',2F7.3)
920   FORMAT(1X,3(4X,D13.6,1X,F7.3))
C
999   RETURN
      END
```

```
      SUBROUTINE FIELDS
C
C
C     THIS SUBROUTINE CALCULATES MODE SUMS FOR THREE ORTHOGONAL
C     FIELD COMPONENTS - EX, EY, AND EZ.
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMPLEX*16 IM/(0.0D0,1.0D0)/,TERM,SUM(3,6),THETA,EXCIT
      REAL*8 IL,IA
      REAL*4 AZIM,CODIP,MAGFLD,SIGMA,EPSR,DIST,AMP,FLD
      CHARACTER*14 LABEL
C
      COMMON/ONE/RHOMIN,RHOMAX,DELRHO,MODES,IPLOT
      COMMON/TWO/AZIM,CODIP,MAGFLD,SIGMA,EPSR
      COMMON/THREE/THETA(10),EXCIT(3,6,10)
      COMMON/FOUR/FREQ,IL,IA,TXHT,RXHT,DIST(500),AMP(6,500),NPTS,
     $  IDPLOT(20),LABEL
      COMMON/FIVE/FLD(3,6,500)
C
      DATA TWOPI/6.283185307D0/,DEGRAD/1.745329D-02/,VELITE/2.997928D2/
      DATA A/6.370D0/,C1/2.857D-3/,C2/5.985D-8/
C
C
      WAVENO = TWOPI*FREQ*1.0D3/VELITE
      CONS1  = C1*IL*DSQRT(FREQ**3)
      CONS2  = C2*IA*DSQRT(FREQ**5)
C
      NPTS = 0
      RHO = RHOMIN
C
C     LOOP OVER RHO
  100 FAC = DSQRT(DSIN(RHO/A))
      NPTS = NPTS+1
      CONST1 = CONS1 /FAC
      CONST2 = CONS2 /FAC
      DIST(NPTS) = RHO
      DO 200 J=1,3
      DO 200 L=1,6
      SUM(J,L) = (0.0D0,0.0D0)
  200 CONTINUE
      DO 400 M=1,MODES
      TERM = CDEXP(-IM*WAVENO*RHO*CDSIN(THETA(M)*DEGRAD))
      DO 300 J=1,3
      DO 300 L=1,6
      SUM(J,L) = SUM(J,L)+EXCIT(J,L,M)*TERM
  300 CONTINUE
  400 CONTINUE
      DO 500 J=1,3
      DO 500 L=1,3
      FLD(J,L,NPTS) = 2.0D1*DLOG10(CDABS(CONST1*SUM(J,L)))
      FLD(J,L+3,NPTS) = 2.0D1*DLOG10(CDABS(CONST2*SUM(J,L+3)))
  500 CONTINUE
      RHO = RHO+DELRHO
      IF(RHO .LE. RHOMAX) GO TO 100
```

```
C
      RETURN
      END
```

```
      SUBROUTINE OUTPUT
C
C
C     THIS SUBROUTINE PRINTS THE FIELDS AND CALLS THE PLOT ROUTINE.
C
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 IL,IA
      REAL*4 DIST,AMP,FLD
      CHARACTER*14 LABEL
      COMMON/ONE/RHOMIN,RHOMAX,DELRHO,MODES,IPLOT
      COMMON/FOUR/FREQ,IL,IA,TXHT,RXHT,DIST(500),AMP(6,500),NPTS,
     $  IDPLOT(20),LABEL
      COMMON/FIVE/FLD(3,6,500)
C
C
      GO TO (740,730,720,710),IPLOT+1
C     EX OPTION
710   LABEL = 'EX - DB/+M*V/M'
      DO 712 L=1,6
      DO 712 N=1,NPTS
      AMP(L,N) = FLD(3,L,N)
712   CONTINUE
      CALL FLDPLT
      PRINT 910
      PRINT 940
      PRINT 950
      DO 714 N=1,NPTS
      PRINT 960,DIST(N),(AMP(L,N),L=1,6)
714   CONTINUE
C     EY OPTION
720   LABEL = 'EY - DB/+M*V/M'
      DO 722 L=1,6
      DO 722 N=1,NPTS
      AMP(L,N) = FLD(2,L,N)
722   CONTINUE
      CALL FLDPLT
      PRINT 920
      PRINT 940
      PRINT 950
      DO 724 N=1,NPTS
      PRINT 960,DIST(N),(AMP(L,N),L=1,6)
724   CONTINUE
C     EZ OPTION
730   LABEL = 'EZ - DB/+M*V/M'
      DO 732 L=1,6
      DO 732 N=1,NPTS
      AMP(L,N) = FLD(1,L,N)
732   CONTINUE
      CALL FLDPLT
740   CONTINUE
      PRINT 930
      PRINT 940
      PRINT 950
      DO 734 N=1,NPTS
```

```
      PRINT 960,DIST(N),(FLD(1,I,N),I=1,6)
  734 CONTINUE
C
  910 FORMAT('0X COMPONENT')
  920 FORMAT('0Y COMPONENT')
  930 FORMAT('0Z COMPONENT')
  940 FORMAT('0',12X,3('ELECTRIC DIPOLE',5X),3('MAGNETIC DIPOLE',5X))
  950 FORMAT( 9X,2(8X,'VERTICAL', 6X,'HORIZONTAL-BROADSIDE',2X,
     $  'HORIZONTAL-ENDON'))
  960 FORMAT(' ',F6.3,6D20.7)
      RETURN
      END
```

```
      SUBROUTINE FLDPLT
C
C
C     USING THE DISSPLA PLOTTING PACKAGE THIS SUBROUTINE GENERATES
C      SIX PLOTS ON ONE 8.5 X 11 PIECE OF PAPER.
C     THE SIX PLOTS ARE:
C        ELECTRIC DIPOLE - VERTICAL
C        ELECTRIC DIPOLE - HORIZONTAL-BROADSIDE
C        ELECTRIC DIPOLE - HORIZONTAL-ENDON
C        MAGNETIC DIPOLE - VERTICAL
C        MAGNETIC DIPOLE - HORIZONTAL-BROADSIDE
C        MAGNETIC DIPOLE - HORIZONTAL-ENDON
C
      REAL*8 FREQ,IL,IA,TXHT,RXHT
      REAL*4 AZIM,CODIP,MAGFLD,SIGMA,EPSR
      CHARACTER*14 LABEL
      DIMENSION Y(500)
      COMMON/TWO/AZIM,CODIP,MAGFLD,SIGMA,EPSR
      COMMON/FOUR/FREQ,IL,IA,TXHT,RXHT,DIST(500),AMP(6,500),NPTS,
     $  IDPLOT(20),LABEL
      DIMENSION PXOR(6),PYOR(6)
      DATA PXOR/3*2.0,3*5.0/
      DATA PYOR/7.0,4.0,1.0,7.0,4.0,1.0/
C
      CALL BGNPL(1)
      CALL INTAXS
      DO 700 J=1,6
      CALL PHYSOR(PXOR(J),PYOR(J))
      DO 600 K=1,NPTS
      Y(K) = AMP(J,K)
  600 CONTINUE
      YMIN = Y(1)
      YMAX = Y(1)
      DO 610 K=2,NPTS
      IF(Y(K) .LT. YMIN) YMIN = Y(K)
      IF(Y(K) .GT. YMAX) YMAX = Y(K)
  610 CONTINUE
      CALL AXSPLT(0.,10.,2.,XOR,XSTEP,XAXIS)
      CALL AXSPLT(YMIN,YMAX,2.,YOR,YSTEP,YAXIS)
      CALL MX1ALF('STAND','*')
      CALL MX2ALF('L/CGREEK','+')
      CALL TITLE(' ',1,'RHO(MM)',7,LABEL,14,XAXIS,YAXIS)
      CALL GRAPH(XOR,XSTEP,YOR,YSTEP)
      CALL CURVE(DIST,Y,NPTS,0)
      IF(J .EQ. 1) CALL MESSAG('ED - V',6,0.6,2.0)
      IF(J .EQ. 2) CALL MESSAG('ED - HB',7,0.6,2.0)
      IF(J .EQ. 3) CALL MESSAG('ED - HE',7,0.6,2.0)
      IF(J .EQ. 4) CALL MESSAG('MD - V',6,0.6,2.0)
      IF(J .EQ. 5) CALL MESSAG('MD - HB',7,0.6,2.0)
      IF(J .EQ. 6) CALL MESSAG('MD - HE',7,0.6,2.0)
      IF(J .EQ. 6) THEN
         CALL MESSAG(IDPLOT,40,-3.0,9.5)
         CALL MESSAG('FREQ=       KHZ',15,-3.0,9.2)
         CALL REALNO(SNGL(FREQ),3,-2.5,9.2)
```

```
      CALL MESSAG('TXHT=        KM',13,-1.0,9.2)
      CALL MESSAG('RXHT=        KM',13, 1.0,9.2)
      CALL REALNO(SNGL(TXHT),1,-0.5,9.2)
      CALL REALNO(SNGL(RXHT),1, 1.5,9.2)
      CALL MESSAG('AZIM=        DEGREES   CODIP=       DEGREES',
     $            40,-3.0,9.0)
      CALL REALNO(AZIM,1,-2.5,9.0)
      CALL REALNO(CODIP,1,0.2,9.0)
      CALL MESSAG('MAGFLD=           W/M   SIGMA=           S/M EPSR=
     $            ,43,-3.0,8.75)
      CALL HEIGHT(.07)
      CALL MESSAG('2',1,-0.7,8.9)
      CALL RESET('HEIGHT')
      CALL REALNO(MAGFLD,-2,-2.3,8.75)
      CALL REALNO(SIGMA,-2,0.2,8.75)
      CALL REALNO(EPSR,1,2.4,8.75)
      CALL MESSAG('IL=          A-M',15,-3.0,8.5)
      CALL MESSAG('IA=          A-M',15,0.0,8.5)
      CALL HEIGHT(.07)
      CALL MESSAG('2',1, 1.8,8.65)
      CALL RESET('HEIGHT')
      CALL REALNO(SNGL(IL),-2,-2.7,8.5)
      CALL REALNO(SNGL(IA),-2, 0.3,8.5)
    ENDIF
    CALL ENDGR(0)
700 CONTINUE
    CALL ENDPL(-1)
C
    RETURN
    END
```

DISTRIBUTION LIST

DEPARTMENT OF DEFENSE
ASSISTANT SECRETARY OF DEFENSE
CMD, CONT COMM & INTELL
DEPARTMENT OF DEFENSE
WASHINGTON, DC 20301
M EPSTEIN
J BABCOCK

DIRECTOR
COMMAND CONTROL TECHNICAL CENTER
11440 ISAAC NEWTON SQUARE, N
RESTON VA 22091
C-650

DIRECTOR
COMMAND CONTROL TECHNICAL CENTER
ROOM ME682, THE PENTAGON
WASHINGTON, DC 20301
C-312

DIRECTOR
DEFENSE ADVANCED RESEARCH PROJECT
AGENCY
1400 WILSON BLVD
ARLINGTON, VA 22209
NUCLEAR MONITORING RSCH
STRATEGIC TECH OFFICE

DEFENSE COMMUNICATION ENGINEERING CENTER
1860 WIEHLE AVENUE
RESTON, VA 22090
CODE R220 (M HOROWITZ)
CODE R720 (JOHN WORTHINGTON)
CODE R410 (JAMES W McLEAN)
CODE R103

DIRECTOR
DEFENSE COMMUNICATIONS AGENCY
WASHINGTON, DC 20305
CODE 810 (RW ROSTRON)
CODE 480
CODE 101B (MAJ ROOD)

DEFENSE COMMUNICATIONS AGENCY
WWMCCS SYSTEM ENGINEERING ORG
WASHINGTON, DC 20305
RL CRAWFORD

DEFENSE TECHNICAL INFORMATION CENTER
CAMERON STATION
ALEXANDRIA, VA 22314
TC (12)

DIRECTOR
DEFENSE INTELLIGENCE AGENCY
WASHINGTON, DC 20301
DIAST-5
DB-4C (EDWARD OFARREL)

DIRECTOR
DEFENSE NUCLEAR AGENCY
WASHINGTON, DC 20305
DDST
TITL TECH LIBRARY (3)
RAAE (3)
STVL

COMMANDER
FIELD COMMAND
DEFENSE NUCLEAR AGENCY
KIRTLAND AFB, NM 87115
FCPR

DIRECTOR
INTERSERVICE NUCLEAR WEAPONS SCHOOL
KIRTLAND AFB, NM 87115
DOCUMENT CONTROL

CHIEF
LIVERMORE DIVISION FLD COMMAND DNA
LAWRENCE LIVERMORE LABORATORY
PO BOX 808
LIVERMORE, CA 94550
FCPRL

DIRECTOR
NATIONAL SECURITY AGENCY
FT GEORGE G MEADE, MD 20755
W65
OLIVER H BARTLETT W32
TECHNICAL LIBRARY
JOHN SKILLMAN R52

OJCS/J-3
THE PENTAGON
WASHINGTON, DC 20301
OPERATIONS (WWMCCS EVAL
OFF, MR TOMA)

OJCS/J-5
THE PENTAGON
WASHINGTON, DC 20301
PLANS & POLICY (NUCLEAR DIVISION)

UNDER SECRY OF DEFENSE FOR RESEARCH
AND ENGINEERING
DEPARTMENT OF DEFENSE
WASHINGTON, DC 2-301
S&SS (OS)

DEPARTMENT OF THE ARMY
COMMANDER/DIRECTOR
ATMOSPHERIC SCIENCES LABORATORY
US ARMY ELECTRONICS COMMAND
WHITE SANDS MISSLE RANGE, NM 88002
DELAS-AE-M (FE NILES)

COMMANDER
HARRY DIAMOND LABORATORIES
2800 POWDER MILL RD
ADELPHI, MD 20783
DELHD-NP (FRANCIES N WIMENITZ)
MILDRED H WEINER DRXDO-II

COMMANDER
US ARMY ELECTRONICS RESEARCH &
DEVELOPMENT COMMAND
FORT MONMOUTH, NJ 07703
DRSEL-RD
(JE QUIGLEY)

COMMANDER
US ARMY FOREIGN SCIENCE & TECH CENTER
220 7TH STREET NE
CHARLOTTESVILLE, VA 22901
R JONES
PA CROWLEY

COMMANDER
US ARMY NUCLEAR AGENCY
7500 BACKLICK ROAD
BUILDING 2073
SPRINGFIELD VA 22150
MONA-WE (J BERBERET)

CHIEF
US ARMY RESEARCH OFFICE
PO BOX 12211
TRIANGLE PARK, NC 27709
DRXRD-ZC

DEPARTMENT OF THE NAVY
CHIEF OF NAVAL OPERATIONS
NAVY DEPARTMENT
WASHINGTON, DC 20350
OP 941
OP-604C3
OP943 (LCDR HUFF)
OP 981

CHIEF OF NAVAL RESEARCH
NAVY DEPARTMENT
ARLINGTON, VA 22217
CODE 402
CODE 420
CODE 421
CODE 461
CODE 464

COMMANDING OFFICER
NAVAL INTELLIGENCE SUPPORT CENTER
4301 SUITLAND RD BLDG 5
WASHINGTON, DC 20390

COMMANDER
NAVAL OCEAN SYSTEMS CENTER
SAN DIEGO CA 92152
CODE 532 (JH RICHTER)
CODE 532 (RICHARD A PAPPART)
CODE 532 (JERRY F FERGUSON)
CODE 5324 (LR HITNEY)

COMMANDING OFFICER
NAVAL RESEARCH LABORATORY
WASHINGTON, DC 20375
CODE 5410 (JOHN DAVIS)
CODE 7701 (JACK D BROWN)
CODE 5461 TRANS IONO PROP
CODE 5465 PROP APPLICATIONS
CODE 5460 ELECTROMAG PROP BR
CODE 2600 TECH LIBRARY (2)

OFFICER-IN-CHARGE
WHITE OAK LABORATORY
NAVAL SURFACE WEAPONS CENTER
SILVER SPRING, MD 20910
CODE WA501 NAVY NUC PRGMS OFF
CODE WX21 TECH LIBRARY

COMMANDER
NAVAL TELECOMMUNICATION COMMAND
NAVTELCOM HEADQUARTERS
4401 MASSACHUSETTS AVE, NW
WASHINGTON, DC 20390
CODE 24C

COMMANDING OFFICER
NAVY UNDERWATER SOUND LABORATORY
FORT TRUMBULL
NEW LONDON, CT 06320
PETER BANNISTER
DA MILLER

DIRECTOR
STRATEGIC SYSTEMS PROJECT OFFICE
NAVY DEPARTMENT
WASHINGTON, DC 20376
NSP-2141

DEPARTMENT OF THE AIRFORCE
COMMANDER
ADC DC
ENT AFB, CO 80912
DC (MR LONG)

COMMANDER
ADCOM/XPD
ENT AFB, CO 80912
XPODQ
XP

AF GEOPHYSICS LABORATORY, AFSC
HASCOM AFB, MA 01731
CRU (S HOROWITZ)
PHP (JULES AARONS)
OPR (JAMES C ULWICK)
OPR (ALVA T STAIR)
SUOL (RESEARCH LIBRARY) (2)

AF WEAPONS LABORATORY, AFSC
KIRTLAND AFB, NM 87117
SUL (2)
SAS (JOHN M KAMM)
DYC (CAPT L WITTWER)

AFTAC
PATRICK AFB, FL 32925
TN
TD-3
TD-5
TF/MAJ WILEY

AIR FORCE AVIONICS LABORATORY, AFSC
WRIGHT PATTERSON AFB, OH 45433
AAD

COMMANDER
FOREIGN TECHNOLOGY DIVISION, AFSC
WRIGHT-PATTERSON AFB, OH 45433
ETD BL BALLARD

HQ USAF RD
WASHINGTON, DC 20330
RDQ

COMMANDER
ROME AIR DEVELOPMENT CENTER, AFSC
GRIFFISS AFB, NY 13440
EMTLD DOC LIBRARY

COMMANDER
ROME AIR DEVELOPMENT CENTER, AFSC
HANSCOM AFB, MA 01731
EEP JOHN RASMUSSEN

SAMSO/MN
NORTON AFB, CA 92409
MINUTEMAN (NMML LTC KENNEDY)

COMMANDER IN CHIEF
STRATEGIC AIR COMMAND
OFFUTT AFB, NB 68113
NRT
XPFS (MAJ BRIAN G STEPHAN)
DOK (CHIEF SCIENTIST)

US ENERGY RESEARCH AND DEV ADMIN
DEPARTMENT OF ENERGY
ALBUQUERQUE OPERATIONS OFFFICE
PO BOX 5400
ALBUQUERQUE, NM 87115
DOC CON FOR DW SHERWOOD

DIVISION OF MILITARY APPLICATION
DEPARTMENT OF ENERGY
WASHINGTON, DC 20545
DOC CON FOR DONALD I GALE

LAWRENCE LIVERMORE NATIONAL
LABORATORY
PO BOX 808
LIVERMORE, CA 94550
GLENN C WERTH L-216
TECH INFO DEPT L-3

LOS ALAMOS NATIONAL SCIENTIFIC
LABORATORY
PO BOX 1663
LOS ALAMOS, NM 87545
DOC CON FOR TF TASCHEK
DOC CON FOR DR WESTERVELT
DOC CON FOR PW KEATON
DOC CON FOR JH COON

SANDIA LABORATORY
LIVERMORE NATIONAL LABORATORY
PO BOX 969
LIVERMORE, CA 94550
DOC CON FOR BE MURPHEY
DOC CON FOR TB COOK ORG 8000

SANDIA NATIONAL LABORATORY
PO BOX 5800
ALBUQUERQUE, NM 87115
DOC CON FOR SPACE PROJ DIV
DOC CON FOR AD THORNBROUGH, ORG 1245
DOC CON FOR WC MYRA
DOC CON FOR 3141 SANDIA RPT COLL

OTHER GOVERNMENT
DEPARTMENT OF COMMERCE
NATIONAL BUREAU OF STANDARDS
WASHINGTON, DC 20234
RAYMOND T MOORE

DEPARTMENT OF COMMERCE
OFFICE OF TELECOMMUNICATIONS
INSTITUTE FOR TELCOM SCIENCE
BOULDER, CO 80302
WILLIAM F UTLAUT
LA BERRY
A GLENN JEAN
DD CROMBIE
JR WAIT

DEPARTMENT OF TRANSPORTATION
OFFICE OF THE SECRETARY
TAD-44.1, ROOM 10402-B
400 7TH STREET, SW
WASHINGTON, DC 20590
RL LEWIS
RH DOHERTY

DEPARTMENT OF DEFENSE CONTRACTORS
AEROSPACE CORPORATION
PO BOX 92957
LOS ANGELES, CA 90009
IRVING M GARFUNKEL

ANALYTICAL SYSTEMS ENGINEERING CORP
5 OLD CONCORD RD
BURLINGTON, MA 01803
RADIO SCIENCES

THE BOEING COMPANY
PO BOX 3707
SEATTLE, WA 98124
GLENN A HALL
JF KENNEY

CUMPUTER SCIENCES CORPORATION
PO BOX 530
6565 ARLINGTON BLVD
FALLS CHURCH, VA 22046
D BLUMBERG

UNIVERSITY OF DENVER
COLORADO SEMINARY
DENVER RESEARCH INTSITUTE
PO BOX 10127
DENVER, CO 80210
DONALD DUBBERT
HERBERT REND

ESL, INC
495 JAVA DRIVE
SUNNYVALE, CA 94086
JAMES MARSHALL

GENERAL ELECTRIC COMPANY
SPACE DIVISION
VALLEY FORGE SPACE CENTER
GODDARD BLVD KING OF PRUSSIA
PO BOX 8555
PHILADELPHIA, PA 19101
SPACE SCIENCE LAB (MH BORTNER)

GENERAL ELECTRIC COMPANY
TEMPO-CENTER FOR ADVANCED STUDIES
816 STATE STREET
PO DRAWER QQ
SANTA BARBARA, CA 93102
B GAMBILL
DASIAC (2)
WARREN S KNAPP

GEOPHYSICAL INSTITUTE
UNIVERSITY OF ALASKA
FAIRBANKS, AK 99701
TN DAVIS
NEAL BROWN
TECHNICAL LABORATORY

GTE SYLVANIA, INC
ELECTRONICS SYSTEMS GRP
EASTERN DIVISION
77 A STREET
NEEDHAM, MA
MARSHAL CROSS

IIT RESEARCH INSTITUTE
10 WEST 35TH STREET
CHICAGO, IL 60616
TECHNICAL LIBRARY

UNIVERSITY OF ILLINOIS
DEPARTMENT OF ELECTRICAL ENGINEERING
URBANA, IL 61803
AERONOMY LABORATORY (2)

JOHNS HOPKINS UNIVERSITY
APPLIED PHYSICS LABORATORY
JOHNS HOPKINS ROAD
LAUREL, MD 20810
J NEWLAND
PT KOMISKE

LOCKHEED MISSILES & SPACE CO, INC
3251 HANOVER STREET
PALO ALTO, CA 94304
EE GAINES
WL IMHOF D/52-12
JB REAGAN D652-12
RG JOHNSON D/52-12

LOWELL RESEARCH FOUNDATION
450 AIKEN STREET
LOWELL, MA 01854
DR BIBL

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LINCOLN LABORATORY
PO BOX 73
LEXINGTON, MA 02173
DAVE WHITE
JH PANNELL L-246
DM TOWLE

MISSION RESEARCH CORPORATION
735 STATE STREET
SANTA BARBARA, CA 93101
R HENDRICK
F FAJEN
M SCHEIBE
J GILBERT
CL LONGMIRE

MITRE CORPORATION
PO BOX 208
BEDFORD, MA 01730
G HARDING

PACIFIC-SIERRA RESEARCH CORP
1456 CLOVERFIELD BLVD
SANTA MONICA, CA 90404
EC FIELD, JR

PENNSYLVANIA STATE UNIVERSITY
IONOSPHERIC RESEARCH LABORATORY
318 ELECTRICAL ENGINEERING EAST
UNIVERSITY PARK, PA 16802
IONOSPHERIC RSCH LAB (2)

R&D ASSOCIATES
PO BOX 9695
MARINA DEL REY, CA 90291
FORREST GILMORE
WILLIAM J KARZAS
PHYLLIS GREIFINGER
CARL GREIFINGER
A ORY
BRYAN GABBARD
RP TURCO
SAUL ALTSCHULER

RAND CORPORATION
1700 MAIN STREET
SANTA MONICA, CA 90406
TECHNICAL LIBRARY (2)
CULLEN CRAIN

SRI INTERNATIONAL
333 RAVENSWOOD AVENUE
MENLO PARK, CA 94025
DONALD NEILSON
GEORGE CARPENTER
WG CHETNUT
JR PETERSON
GARY PRICE

STANFORD UNIVERSITY
RADIO SCIENCE LABORATORY
STANFORD, CA 94305
RA HELLIWELL
FRASER-SMITH
J KATSUFRAKIS

TRW DEFENSE & SPACE SYS GROUP
ONE SPACE PARK
REDONDO BEACH, CA 90278
DIANA DEE

CALIFORNIA INSTITUTE OF TECHNOLOGY
JET PROPULSION LABORATORY
4800 OAK GROVE DRIVE
PASADENA, CA 91103
ERNEST K SMITH
(MAIL CODE 144-B13)

END

DTIC

8-86